PCT/AU2004/000312

REC'D 06 APR 2004

WIPO PCT

Patent Office
Canberra

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

I, JULIE BILLINGSLEY, TEAM LEADER EXAMINATION SUPPORT AND SALES hereby certify that annexed is a true copy of the Provisional specification in connection with Application No. 2003901237 for a patent by THE AUSTRALIAN NATIONAL UNIVERSITY as filed on 14 March 2003.

WITNESS my hand this
Thirtieth day of March 2004

J. Billingsley

JULIE BILLINGSLEY
TEAM LEADER EXAMINATION
SUPPORT AND SALES

BEST AVAILABLE COPY

Regulation 3.2

AUSTRALIA

Patents Act 1990

PROVISIONAL SPECIFICATION

for the invention entitled:

"Fractal Image Data and Image Generator"

The invention is described in the following statement:

FRACTAL IMAGE DATA AND IMAGE GENERATOR

FIELD OF THE INVENTION

The present invention relates to fractals, and in particular both to a new class of fractals and also to a fractal generator able to iteratively generate image data and a number of such fractals.

BACKGROUND

Since the 1980's, fractals have been used to describe a wide variety of physical systems. Fractals can be generally classed as deterministic, if they can be generated by a deterministic process, or random, if the generation process uses one or more probability distributions to select iterated function systems. One of the best known applications of fractals is the simulation of natural formations occurring in the physical world, such as mountains, clouds, and waves. One of the difficulties of existing fractals and procedures for their generation is that deterministic fractals tend to be too rigidly deterministic, and random fractals too random, when these are compared with natural processes and forms occurring in the physical world. Deterministic fractals, while initially eye-catching, tend to lose the viewer's interest because they appear too repetitive. On the other hand, random fractals can also appear somewhat boring because the eye finds it hard to discern patterns, similar to the way that wispy white clouds can appear boring. These weaknesses are one of the factors that have limited the use of fractals in the content industries, in the areas of special effects in digital animation, computer games, and advertising, for example.

It is desired to provide at least a useful alternative, and in particular to provide image data representing at least a nascent fractal and a generator and process for generating the image data that alleviate one or more of the above difficulties.

SUMMARY OF THE INVENTION

In accordance with the present invention, there is provided a V-variable fractal

In particular, the present invention provides a V-variable fractal represented by fractal image data, where V is an integer greater than one and represents the number of constituent images available for iterative combination to generate the fractal. The images are combined in a random manner.

The present invention also provides image data, wherein a variable number n of constituent images are iteratively transformed and combined in a random manner to generate the image data, with $1 < n \leq V$.

Preferably, the constituent images are transformed using projective transformations at each iteration.

The present invention also provides a fractal generation process, including:

(i) randomly selecting images from a set of input images;

(ii) selecting transformation functions from a set of transformation functions;

(iii) generating transformed images by applying the selected transformation functions to the selected images; and

(iv) generating an output image by combining the transformed images.

Preferably, the process includes:

(v) repeating steps (i) to (iv) to generate a set of output images; and

(vi) repeating steps (i) to (v) using said set of output images as said set of input images to generate a new set of output images.

Preferably, the process includes repeating step (vi) until said new set of output images is substantially independent of the first set of input images used in the process.

Preferably, the number of selected transformation functions is less than the number of transformation functions in said set of transformation functions.

Preferably, the step of selecting transformation functions includes selecting an iterated function system from a set of iterated function systems, each iterated function system including a set of transformation functions.

Preferably, the selection of an iterated function system is based on selection probabilities associated with said iterated function systems.

Advantageously, the combining of said transformed images may include superimposing said transformed images.

Advantageously, said transformation functions may include geometrical transformations.

Preferably, said geometrical transformations include scaling and translation.

Advantageously, said geometrical transformations may include scaling, translation and geometrical distortion.

Preferably, said transformation functions are contractive on average.

Advantageously, said geometrical transformations may be contractive transformations.

Advantageously, said transformation functions may include projective transformations.

Advantageously, said transformation functions may include transformations of brightness and/or colour.

Advantageously, each of said transformation functions may be represented by one or more parameters.

Advantageously, the process may include generating said transformation functions and said selection probabilities.

Advantageously, said transformation functions and said selection probabilities may be generated on the basis of predetermined probability distributions.

The present invention also provides a fractal generation process, including randomly selecting from a set of input images, transforming the selected images, and combining the transformed images to generate a set of output images, and iterative repetition of these steps using the set of output images of each iteration as the set of input images for the next iteration.

Advantageously, said selecting may include selecting the same input image more than once.

Preferably, said transforming includes scaling and translating the selected images.

Advantageously, said transforming may also include geometrically distorting the selected images.

Preferably, the transforming is contractive.

The present invention also provides a fractal generator, including:

an image selector for selecting M of V input images;

a function selector for selecting a set of M transformation functions;

at least one image transformer for respectively applying the selected transformation functions to the selected input images; and

a compositor for composing an output image from the images output by said at least one image transformer.

Advantageously, the fractal generator may include a set of V image buffers storing the input images.

Preferably said function selector selects said set from N sets of transformation functions.

The present invention also provides a system having components for executing the steps of any one of the above processes.

The present invention also provides program code for executing the steps of any one of the above processes.

The present invention also provides a computer readable storage medium having stored thereon program code for executing the steps of any one of the above processes.

The present invention also provides fractal data generated by any one of the above processes.

The said fractal data represents one or more V-variable fractals.

The present invention also provides a fractal generation system, including an image selector for selecting images from a set of input images, and an image transformer for transforming the selected images to generate a set of output images, said system being adapted to provide said set of output images as the set of input images to iteratively generate fractal image data.

Preferably, said image transformer includes one or more image transformation modules for transforming said selected images, and an image combination module for combining the transformed images.

Preferably, said one or more image transformation modules are adapted to scale and translate said selected images.

Advantageously, said one or more image transformation modules may also be adapted to geometrically distort said selected images.

Preferably, the system includes a transformation selection module to select transformations to be applied to the selected images from a set of transformations.

Preferably, said transformation selection module is adapted to select transformations based on selection probabilities associated with said set of transformations.

The present invention also provides fractal image data representing a combination of two or more constituent first images, each of said first images representing a random transformed combination of two or more constituent second images, each of said second images representing a random transformed combination of two or more constituent third images, each of said third images representing a random transformed combination of two or more constituent fourth images, wherein each transformation includes at least one of translation and rotation.

Preferably, each transformation is a projective transformation, such as an affine transformation.

Preferably, each transformation includes contractive scaling.

Preferably, the transformations are contractive on average.

Advantageously, each combination may be a superposition.

BRIEF DESCRIPTION OF THE DRAWINGS

Preferred embodiments of the present invention are hereinafter described, by way of example only, with reference to the accompanying drawings, wherein:

Figure 1 is a block diagram of a preferred embodiment of a fractal generation system;

Figure 2 is a block diagram of a fractal generator of the fractal generation system;

Figure 3 is a block diagram of an image transformer of the fractal generator;

Figure 4 is a flow diagram of a fractal generation process executed by the fractal generation system;

Figure 5 is a schematic diagram illustrating transformation of an input image by a transformation module of the image transformer;

Figures 6 to 10 are screenshots of images generated by the system using a first set of input parameters;

Figures 11 to 15 are screenshots of images generated by the system using a second set of input parameters; and

Figures 16 to 23 are screenshots of images generated by the system using a third set of input parameters.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

As shown in Figure 1, a fractal generation system includes a fractal generator 100. As shown in Figure 2, the fractal generator 100 includes an image selector 202, an iterated function system (IFS) selector 204, and an image transformer 206. The fractal generation system executes a fractal generation process that generates output image data 102 from input image data 104, iterated function systems 106, and selection probabilities 108. The input image data 104 constitutes a set of input images 104, and the output image data 102 constitutes two or more output images 102 that may be considered as representing two or more fractals, as described below.

In the described embodiment, the fractal generation system is a standard computer system such as an Intel™ x86-based personal computer including a Pentium™ processor 110,

random access memory (RAM) 112, and non-volatile (*e.g.*, magnetic disk) storage 114, and the fractal generator is implemented by software modules stored on the non-volatile storage 114 of the computer and executed by the processor 110. However, it will be apparent that at least parts of the fractal generator can be alternatively implemented by dedicated hardware components, such as application-specific integrated circuits (ASICs).

The fractal generation system is described as generating output image data 102 from input image data 104. Although it is not necessary that the input and output data relate to images as such, *i.e.*, the system processes these as arbitrary numeric data, it is expected that the system will be used to process input data and generate output data that in each case will be visualised as one or more images on a display or hardcopy. Accordingly, the input image data 104 is hereinafter referred to as input images 104, and the output image data 102 is hereinafter referred to as output images 102.

A user of the fractal generation system provides to the system a number $V > 1$ (*i.e.*, at least two) input images 104, together with $N > 1$ iterated function systems $F = \{F^1, \ldots, F^N\}$ 106 and N selection probabilities $P = \{P^1, \ldots, P^N\}$ 108. Each of the N iterated function systems 106 comprises a set of M transformation functions. Each transformation function defines a contractive affine transformation that can be applied to an image, and is represented by the six parameters r, s, x, y, θ and φ. Other transformations can be used, each represented by a set of parameters. As shown in Figure 5, an input image 502 (in this case having horizontal and vertical dimensions of unity) is transformed by first scaling the horizontal and vertical dimensions of the input image 502 by the factors r and s, respectively, with $-1 < r < 1$ and $-1 < s < 1$, followed by translation along the x and y axes by the amounts specified by the x and y parameters, respectively, followed by a geometrical distortion whereby horizontal lines are rotated by the angle θ, and vertical lines are rotated by the angle φ, with $-180 \le \theta \le 180$ and $-180 \le \varphi \le 180$. In Figure 5, the angles θ and φ, are nearly equal, and the output images 504 appears to retain its rectangular shape. However, when these two parameters are substantially unequal, a rectangular input image 502 is distorted to a generally parallelogram or diamond-like shape.

Because each transformation function is represented by the six transformation parameters described above, each iterated function system (IFS) 106, being a set of M transformation functions, can be represented as a parameter table 506 with M rows, each row representing a distinct transformation function. For example, an IFS 106 for M=3 can be represented as a 3×6 matrix or table, as follows:

r_1	s_1	x_1	y_1	θ_1	φ_1
r_2	s_2	x_2	y_2	θ_2	φ_2
r_3	s_3	x_3	y_3	θ_3	φ_3

Thus the iterated function systems 106 are provided as a set of $N > 1$ parameter tables, each having $M > 1$ rows. The N selection probabilities 108 are used to select a particular IFS (*i.e.*, parameter table) to apply to selected images, as described below.

As shown in Figure 2, the fractal generator 100 operates on a set of V input buffers 208 to 212, and a set of V output buffers 214 to 218 allocated from the computer RAM 112. Although Figure 2 shows V=3 of each kind of buffer, there can be in general an arbitrary number $V > 1$ of each buffer type, subject only to system constraints. The number of input images 104 provided to the system is less than or equal to the number V of input and output buffers, as the same image can be allocated to more than one buffer, and the input images 104 are copied into the respective input buffers 208 to 212. The fractal generator 100 then executes a fractal generation process, as shown in Figure 4. The fractal generator 100 generates output images into each of the output buffers 214 to 218 in a sequential manner, starting with the first output buffer 214, and ending with the last output buffer 218. When the outermost loop of the fractal generation process is executed, an iteration of the process is said to have completed, and occurs when all the output buffers 214 to 218 have images stored in them.

The fractal generation process begins at step 402, when the first output buffer 214 is selected to receive the first output image. At step 404, the image selector 202 randomly

selects M of the V input images stored in the input buffers 208 to 212, with the same image allowed to be selected more than once. Accordingly, each time a selection is made, the probability of selecting the input image from a particular buffer is equal to 1/V, although it will be apparent that alternative selection methods can be used. The M selected images are provided to the image transformer 206. At step 406, the IFS selector 204 randomly selects one of the N IFSs 106 using the respective N selection probabilities 108. For example, the N selection probabilities 108 can be provided as respective weights used to divide the numeric interval between 0 and 1 into N contiguous subdivisions, where the size of each subdivision is proportional to the weight for that subdivision. A random number between 0 and 1 can then be generated to select one of the N IFSs 106, according to which subdivision the random number falls within.

As shown in Figure 3, the image transformer 206 includes M transformation modules 302, 304, and a superimpose module 306, shown for the example of M=2. At step 408, the transformation modules 302, 304 transform the selected input images using the selected IFS. The selected input images are transformed using respective transformation functions from the selected IFS. That is, the first transformation module 302 applies a first transformation function defined by the first row 312 of the selected parameter table 316 to the first selected input image 308. Similarly, the second transformation module 304 applies a second transformation function defined by the second row 314 of the selected parameter table 316 to the second selected input image 310.

Each of the transformation modules 302, 304 generates a transformed version of its input image, using the corresponding transformation function, as described above. It will be appreciated that, although distinct transformation modules 302, 304 are shown in Figure 3, the M selected input images could alternatively be processed sequentially by a single transformation module. However, it may be particularly advantageous to provide distinct execution instances of each of the transformation modules 302, 304 if the system includes more than one processor 110 or if the transformation modules 302, 304 are implemented as dedicated hardware components.

In the preferred embodiment, the input images 104 provided to the system by a user are $m \times n$ rectangular images utilising a rectangular co-ordinate system with origin 0,0 corresponding to one of the corners of the rectangular input images 104. The transformed images generated by the transformation modules 302, 304 are maintained as $m \times n$ images corresponding to the same spatial co-ordinates as the original input images 104. Accordingly, a given transformed image will generally include a number of transformed versions of previous images positioned within an otherwise empty $m \times n$ space within the particular transformed image. Any transformed images or parts of transformed images that fall outside the $m \times n$ co-ordinate space of a transformed image are cropped and are not included within the transformed image. Because each transformed image corresponds to the same $m \times n$ region of coordinate space, transformed images can be combined by superposition. However, it will be apparent that alternative embodiments can be devised such that a transformed image is not generated as an $m \times n$ rectangular image within the $m \times n$ coordinate space defined by input images 104, and the combining of images may involve only partial overlap of the constituent images, or may not involve any overlap at all

At step 410, the resulting transformed images are processed by the superimpose module 306. This involves superimposing or overlaying the transformed images to generate a single output image 318. The superimposition is achieved by assigning to each pixel of the output image 318 a colour value equal to the average of the colour values of the corresponding pixels of the transformed images. The resulting superimposed output image 318 is stored in the current output buffer, in this case the first output buffer 214.

It will be apparent that the transformed images can be combined in a variety of alternative ways. For example, the images may be combined by putting one down on top of the preceding one, one after another, so that only the last value written to each pixel is kept. Alternatively, a depth can be associated with each pixel in each image, for example, by adding another dimension to each image and increasing the parameter set from 2D to 3D, and then, when combining images, the frontmost pixel from the images being combined is

the one that is used, as is done in 3D computer graphics, where a Z-buffer is used to select the frontmost pixels. Or alternatively, the exclusive or (XOR) of the first image and the second image can be used.

At step 412, a check is performed to determine whether the current output buffer is the last output buffer 218. If not, then the next output buffer, in this case the second output buffer 216, is selected at step 414, and the process returns to step 404 to randomly select another M images from the V input buffers 208 to 212, and subsequently generate another superimposed output image 318 for storage in the second output buffer 216 by executing steps 406 to 410. These steps are repeated until all the V output buffers 214 to 218 have been used to store a new superimposed output image 318. Once the last output buffer 218 has stored an image, the test at step 412 succeeds, and an iteration of the process is complete. Rather than selecting another output buffer, the contents of the output buffers 214 to 218 are then used as input images to the next iteration of the process. This can be achieved by physically copying the contents of the output buffers 214 to 218 into the respective input buffers 208 to 212. However, it is more efficient simply to exchange, at step 416, the roles of the two sets of buffers by exchanging or swapping the corresponding buffer pointers into the random access memory 112. In any case, after the superimposed output images have been provided in respective input buffers 208 to 212 and the output buffers 214 to 218 have been cleared, the fractal generation process begins the next iteration by selecting the first output buffer 214 at step 402, and repeating the steps described above. As many iterations as desired can be performed.

The contents of one or more of the output buffers 214 to 218 can be displayed on a display device associated with the fractal generation system for viewing by the user. Individual images selected by the user or images generated by a user-specified number of iterations of the fractal generation process can be stored as final output images 102 on the non-volatile storage medium 114 for subsequent use. After a number of iterations, the output images 102 are a set of V-variable fractal images derived from the preceding input images. It will be appreciated by those skilled in the art that the although the image data generated by the fractal generation system is referred to as fractal image data, it may only represent a

nascent fractal if the finite number of iterations of the fractal generation process are not sufficient to produce a fractal.

With successive iterations, the output images 318 become less dependent on the initial input images 104, and eventually become characteristic of the input parameter tables 106 and their selection probabilities 108. The images 318 generated by the system then form a diverse sample of an infinite collection of all possible images associated with the initial parameter tables 106 and probabilities 108. In particular, because the transformations are contractive, the original input images 104 for the first iteration will eventually be contracted to the size of a single pixel when the entire output image is viewed, given a sufficient number of iterations. Thus the output images become independent of the original input images 104. Accordingly, the input images 104 can be alternatively provided or generated by the system itself, rather than being provided by the user as input, and the value of V can be provided as input to the system. By adjusting the parameter tables 106, the appearance and form of the classes of images generated by the system can be modified. For example, one class of images might resemble a group of tea trees, another a collection of faces, and another might resemble (and be used to represent) possible stock market simulations. As a result, the possible applications of the fractal generation system and process are very broad and diverse.

For example, Figures 6 to 10 are screenshots of images generated by the fractal generation system after 4, 5, 6, 11 and 44 iterations, respectively, using V=2 and the following input parameter tables:

IFS 1:

	a	b	c	d	e	f
transformation 1:	0.5	-0.375	0.3125	0.5	0.375	.1875
transformation 2:	0.5	0.375	0.1875	-0.5	0.375	.6875

IFS 2:

	a	b	c	d	e	f
transformation 1:	0.5	-0.375	0.3125	-0.5	-0.375	.8125
transformation 2:	0.5	0.375	0.1875	0.5	-0.375	.3125

In this example, each transformation is an affine transformation defined by (x,y) → (ax+by+c, dx+ey+f), where the parameters a to f are as indicated in the parameter tables.

Figures 11 to 15 are screenshots of fractal images generated after 1, 3, 6, 8, and 22 iterations, respectively, using V=2 and the following parameter tables:

IFS 1:

a	b	c	d	e	f	g	h	I
.151827	.034074	-.215212	-.123706	-.246197	.054262	-.212523	.059302	-.234997
.011248	.011321	-.000425	-.001814	-.026793	.062591	.014512	-.057923	.086631

IFS 2:

a	b	c	d	e	f	g	h	I
.017380	-.035039	.046965	.029178	.006695	.025663	.021377	-.012129	.048469
.144191	.004070	-.183496	-.099432	-.243409	.043246	-.151388	.001065	-.199968

In this example, each transformation is a projective transformation defined by (x,y) → ((ax+by+c) / (gx+hy+I), (dx+ey+f)) / (gx+hy+I)) , where the parameters a to I are as indicated in the parameter tables.

Figures 16 to 23 are screenshots of images generated after 1, 2, 3, 4, 6, 8, 22, and 40 iterations, respectively, using the same projective transformation with V=2 and the following parameter tables:

IFS 1:

a	b	c	d	e	f	g	h	I
.001231	-.335970	.002318	-.164124	-.028044	-.125233	.003548	-.033609	-.309261
.002575	.039210	.007817	-.045553	-.005329	.056137	-.001851	-.027965	.126001

IFS 2:

a	b	c	d	e	f	g	h	I
-.006211	-.197079	.208860	.087110	.013677	.011083	-.008298	.075200	.205408
.003412	.306195	-.303822	.141253	.023639	-.286545	.001890	.028936	-.312419

The image data or images generated by the system represent a new type or class of fractals, referred to as V-variable fractals. As can be seen from Figures 6 to 23, they can be considered to be random fractals because they exhibit a degree of randomness, but this is intermediate between deterministic fractals and prior art random fractals generated by other methods. As such, they are able to represent natural forms more accurately than prior art fractals, and yet can be generated more readily than prior art random fractals.

An image representing a V-variable fractal can be characterised as follows. Each fractal image is a combination of at least two smaller images. There is a set of V basic images such that each of the smaller images represents a contractive affine transformation of a basic image from the set of V basic images. The V basic images and the contractive affine transformations provide the first level of decomposition of the image representing the V-variable fractal. Each of the two or more smaller images from the first level of decomposition is itself a combination of at least two smaller again images and there is a second set of V basic images such that each of the smaller again images represents a contractive affine transformation of a basic image from the second set of basic images. The second set of V basic images and the corresponding contractive affine transformations provide a second level of decomposition of the image representing the V-variable fractal. These decompositions can be continued for at least four levels of decomposition. However, if every set of basic images contains just one distinct basic image, then the representing

image is a deterministic or random fractal of a previously known type, and as such is not included in the new class of fractals.

It will be apparent that a variety of alternative transformations can be used in the above manner. For example, projective transformations other than affine transformations can be used.

Figures 6 to 10, 13 to 15, and 19 to 23 are examples of V-variable fractal images where V=2. Each of these images was generated by four or more iterations of the fractal generation process. At each scale there are at most V non-equivalent sets. These sets can be combined partially or wholly one on another. The sets typically change from magnification level to magnification level. While V=2 in Figures 6 to 23, larger values of V can be used.

The properties of V-variable fractals are described in more detail in M. Barnsley, J. Hutchinson and Ö. Stenflo, *A New Random Iteration Algorithm and its Attractors*, included as Appendix I.

The fractal generation system has been described above in terms of contractive affine geometrical transformation and superimposition of two-dimensional images. However, it will be apparent that the iterated function systems 106 can include other function systems that can be used to modify the input image data 104. For example, the iterated function systems 102 can include contractive projective transformations, or affine transformations that are contractive on average, where the average is determined by the probabilities 108 associated with the iterated function systems 106. Because the system generates output images by combining multiple, transformed copies of input images, contractive transformations are preferably used in order to limit the output image size when translated and/or rotated images are combined, and this is preferably limited to the size of the input images. However, the transformations can alternatively be contractive on average, as described above. Similarly, it will be apparent that the transformed images can be

combined by methods other than superposition, as described above. Moreover, the fractal generation process can be applied to image data representing three-dimensional images to generate a new class of three-dimensional fractals, or indeed to input data of even higher dimensionality. The image data need not represent an existing image or be used to generate an image, but can be used for any desired purpose.

In an alternative embodiment, the transformation modules 302, 304 can modify the colour and/or brightness of input images, in addition to the geometrical transformations described above. These additional modifications are defined by additional parameters included in the parameter tables 106. When the modifications of colour and/or brightness are performed in a contractive manner, the output images generated by the system, after a small number of iterations, depend only on the input parameter tables 106 and their selection probabilities 108. If the brightness of an image is represented by a number z, then an example of a contractive transformation of brightness is new_z= p*z + q where $|p|<1$ and q is a constant. Contractive transformations of colour are achieved by three contractive transformations of brightness, one for each of the red, blue and green colour components.

In a further embodiment, the parameter tables are generated graphically. For example, an affine transformation can be characterised by how it transforms a given triangle into another triangle. For example, a user can use a pointing device (such as a mouse) to drag the vertices of a given triangle to new positions and thereby define an affine transformation.

In yet a further alternative embodiment, the parameter tables 106 and selection probabilities 108 are themselves generated from predetermined probability distributions, such as normal distributions, using a random number generator.

It may be appreciated that the new class of V-variable fractals, realised as images or in any digital form, may also be generated by processes other than those described above. For example, those skilled in the art will realise that for the generation processes described

herein, which correspond to forward methods, there are also corresponding backward methods, although such methods will be less efficient for generating V-variable fractals.

Many modifications will be apparent to those skilled in the art without departing from the scope of the present invention as herein described with reference to the accompanying drawings.

DATED this 14th day of March, 2003

THE AUSTRALIAN NATIONAL UNIVERSITY
By its Patent Attorneys
DAVIES COLLISON CAVE

APPENDIX I

A NEW RANDOM ITERATION ALGORITHM AND ITS ATTRACTORS

MICHAEL BARNSLEY, JOHN HUTCHINSON, AND ÖRJAN STENFLO

ABSTRACT. We describe a new class of random fractals and fractal measures with the remarkable property that they can be computed using a forward process; that is, a version of the Random Iteration Algorithm can be used to sample the random fractals. The results are obtained by constructing a new iterated function system (a super IFS) out of a collection of IFSs with probabilities. The attractor of the super IFS is a collection of sets or measures and associated (probability) measures, (what we call a superfractal.) When the underlying space is for example $\mathbb{R}^2$, and the transformations are computationally straightforward, such as affine transformations, the superfractal can be sampled means of the algorithm, which is not only highly efficient in terms of memory usage but also the fractals that it produces are special; at each magnification level there is a predominance of a few key "forms" or "shapes". The algorithm is illustrated by some computed examples. Some variants, special cases, generalizations of the framework, and potential applications are mentioned.

1. INTRODUCTION AND NOTATION

1.1. Fractals and Random Fractals. A theory of deterministic fractal sets and measures, using a "backward" algorithm, was developed in Hutchinson [5]. A different approach using a "forward" algorithm was developed in Barnsley and Demko [12].

Falconer [16], Graf [17] and Mauldin and Williams [4] randomized each step in an approximation of a "backward" process to obtain random fractal sets. Arbeiter [21] introduced and studied random fractal measures; see also Olsen [23]. ** Insert a sentence on [7] and [8]. For further material see Zähle [25], Patzschke and Zähle [24], and the references in all of these.

Deterministic fractal sets and measures are defined as the attractors of certain iterated function systems (IFSs), see Section 2 for details. Approximations in practical situations quite easily can be computed using the associated random iteration algorithm. Random fractals are typically harder to compute because one has to first calculate lots of fine random detail at low levels, then one level at a time, build up the higher levels.

In this paper we restrict the class of random fractals to ones that we call V-variable fractals. Superfractals are sets of V-variable fractals. They can be defined using a new type of IFS, in fact an IFS made of a finite number N of IFSs, and there

Date: January 14, 2003.
2000 *Mathematics Subject Classification.* Primary xxxx; Secondary xxxx, xxxx.
Key words and phrases. Iterated Function Systems, Stochastic Processcss, Random Fractals.

is available a novel random iteration algorithm: each iteration produces a new V-variable fractal, lying increasingly close to the superfractal, and moving ergodically around it.

Superfractals appear to be a new class of geometrical object, their elements lying somewhere between deterministic fractals which correspond to $V = N = 1$, and random fractals which correspond to $V = \infty$. They seem to allow geometric modelling of some natural objects, examples including realistic-looking leaves, clouds, and textures; and good approximations can be computed fast in elementary computergraphical examples. They are fascinating to watch, one after another, on a computer screen, diverse, yet ordered enough to suggest coherent natural phenomena and potential applications.

Areas of potential applications include computer graphics, stock market simulation, and simulation using finite V of a case with $V = \infty$ corresponding to Brownian motion.

1.2. **An Example.** Here we give an illustration of an application of the theory in this paper. By means of this example we introduce informally V-variable fractals and superfractals. We also explain why we think these objects are of special interest and deserve attention. We start with two pairs of contractive affine transformations, $\{f_1^1, f_2^1\}$ and $\{f_1^2, f_2^2\}$, where $f_m^n : \square \to \square$ with $\square := [0,1] \times [0,1] \subset \mathbb{R}^2$. We use two pairs of screens, where each screen corresponds to a copy of $\square$ and represents for example a computer monitor. We designate one pair of screens to be the Input Screens, denoted by $(\square_1, \square_2)$. The other pair of screens is designated to be the Output Screens, denoted by $(\square_{1'}, \square_{2'})$.

Initialize by placing an image on each of the Input Screens, as illustrated in Figure 2, and clearing both of the Output Screens. We construct an image on each of the two Output Screens as follows.

(i) Pick randomly one of the pairs of functions $\{f_1^1, f_2^1\}$ or $\{f_1^2, f_2^2\}$, say $\{f_1^{n_1}, f_2^{n_1}\}$. Apply $f_1^{n_1}$ to one of the images on $\square_1$ or $\square_2$, selected randomly, to make an image on $\square_{1'}$. Then apply $f_2^{n_1}$ to one of the images on $\square_1$ or $\square_2$, also selected randomly, and overlay the resulting image I on the image now already on $\square_1$. (For example, if black-and-white images are used, simply take the union of the black region of I with the black region on $\square_1$, and put the result back onto $\square_1$.)

(ii)Again pick randomly one of the pairs of functions $\{f_1^1, f_2^1\}$ or $\{f_1^2, f_2^2\}$, say $\{f_1^{n_2}, f_2^{n_2}\}$. Apply $f_1^{n_2}$ to one of the images on $\square_1$, or $\square_2$, selected randomly, to make an image on $\square_{2'}$. Also apply $f_2^{n_2}$ to one of the images on $\square_1$, or $\square_2$, also selected randomly, and overlay the resulting image on the image now already on $\square_{2'}$.

(iii) Switch Input and Output, clear the new Output Screens, and repeat steps (i), (ii) and (iii) many times, to allow the system to settle into some sort of "stationary state".

What sorts of images do we see on the successive pairs of screens, and what sorts of images are produced in the "stationary state"? What sorts of things does the theory developed in this paper tell us about such situations? As a specific example, let us choose

$$f_1^1(x,y) = (0.5x - 0.375y + 0.3125, 0.5x + 0.375y + 0.1875), \tag{1.1}$$

$$f_2^1(x,y) = (0.5x + 0.375y + 0.1875, -0.5x + 0.375y + 0.6875), \tag{1.2}$$

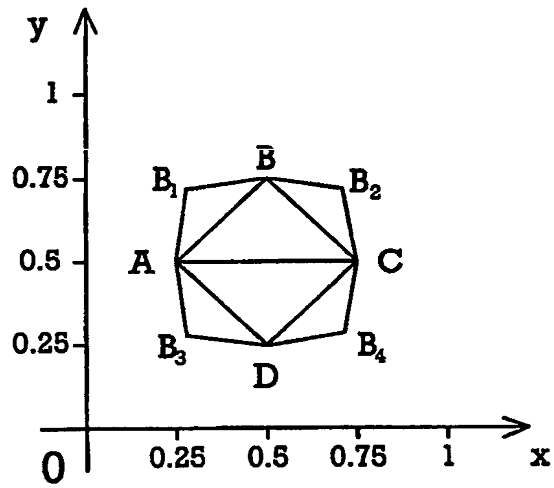


FIGURE 1. Triangles used to define the four transformations f_1^1, f_2^1, f_1^2, and f_2^2.

$$f_1^2(x,y) = (0.5x - 0.375y + 0.3125, -0.5x - 0.375y + 0.8125), \tag{1.3}$$

$$f_2^2(x,y) = (0.5x + 0.375y + 0.1875, 0.5x - 0.375y + 0.3125). \tag{1.4}$$

We describe how these transformations act on the triangle ABC in the diamond $ABCD$, where $A = (0.25, 0.5)$, $B = (0.5, 0.75)$, $C = (0.75, 0.5)$, and $D = (0.5, 0.25)$. Let $B_1 = (0.281\,25, 0.718\,75)$, $B_2 = (0.718\,75, 0.718\,75)$, $B_3 = (0.281\,25, 0.281\,25)$, and $B_4 = (0.718\,75, 0.281\,25)$. See Figure 1 (to be inserted). Then we have

$$f_1^1(A) = A,\ f_1^1(B) = B_1,\ f_1^1(C) = B; \tag{1.5}$$

$$f_2^1(A) = B,\ f_2^1(B) = B_2,\ f_2^1(C) = C; \tag{1.6}$$

$$f_1^2(A) = A,\ f_1^2(B) = B_3,\ f_1^2(C) = D; \tag{1.7}$$

$$f_2^2(A) = D,\ f_2^2(B) = B_4,\ f_2^2(C) = C. \tag{1.8}$$

In Figure 2 we show an intial pair of images, two jumping fish, one on each of the two screens $\square_1$ and $\square_2$. In the subsequent Figures, 3, 4, 5, 6, 7, 8, and 9, we show the start of the sequence of pairs of images actually obtained in a particular trial, for the first seven iterations. Then in Figures 10, 11, and 12, we show three successive pairs of screens, obtained after more than twenty iterations. These latter images are typical of the ones obtained for any number more than twenty iterations, very diverse, but always representing continuous "random" paths in $\mathbb{R}^2$; they correspond to the "stationary state", at the resolution of the images. Notice how the two images in Figure 11 consists of the union of shrunken copies of the images in Figure 10, while the curves in Figure 12 are made from two shruñken copies of the curves in Figure 11.

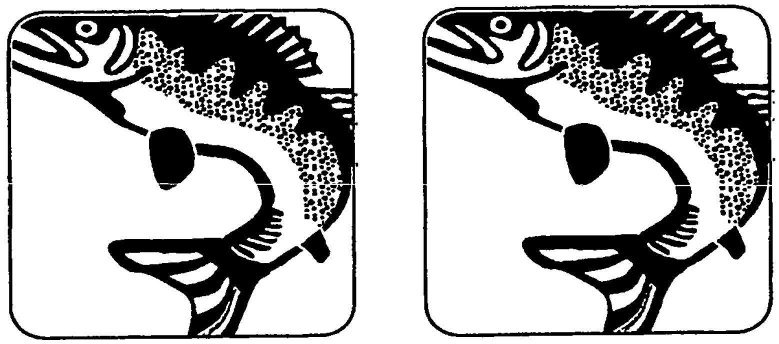


FIGURE 2. An initial image of a jumping fish on each of the two screens $\square_1$ and $\square_2$.

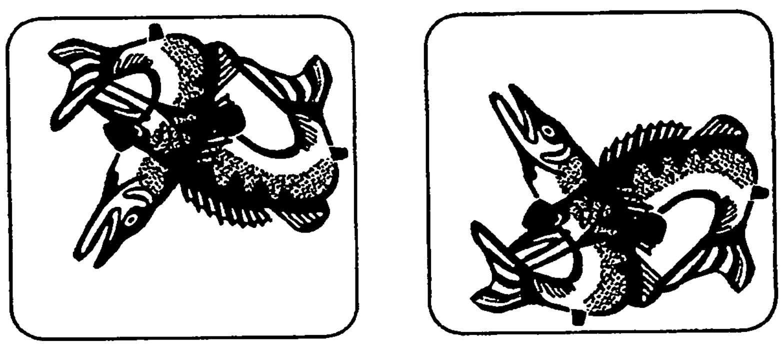


FIGURE 3. The pair of images after one iteration.

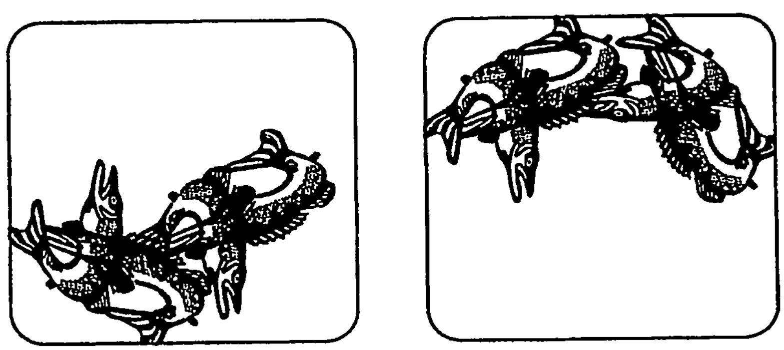


FIGURE 4. The two images after two iterations.

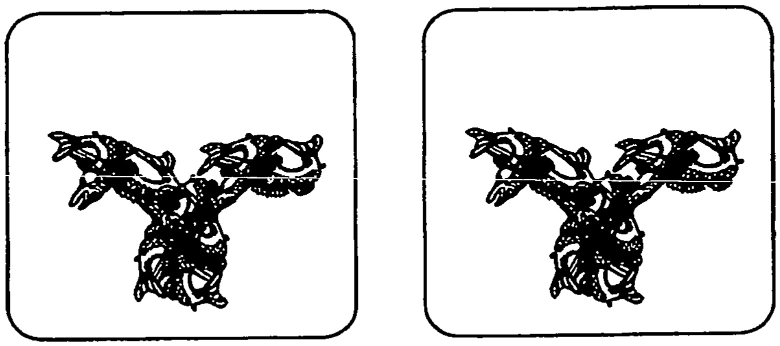


FIGURE 5. The two images after three iterations. Both images are the same.

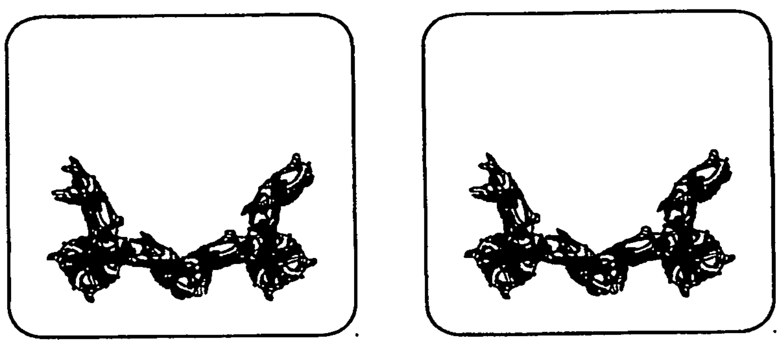


FIGURE 6. The two images after four iterations. Both images are again the same, a braid of fish.

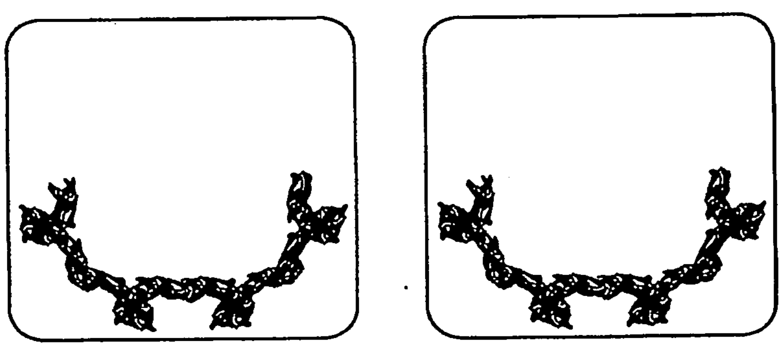


FIGURE 7. The two images after five iterations. The two images are the same.

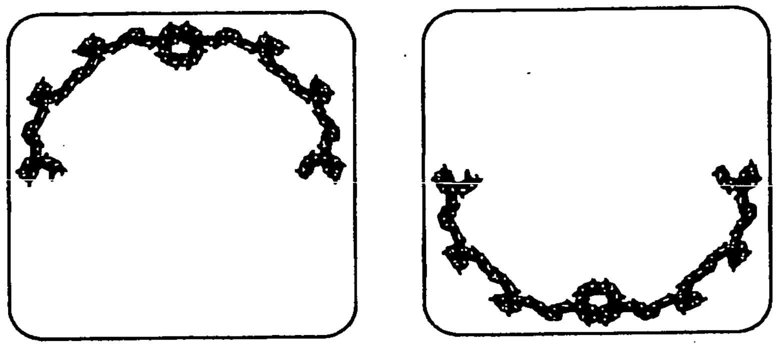


FIGURE 8. The two images after six iterations.

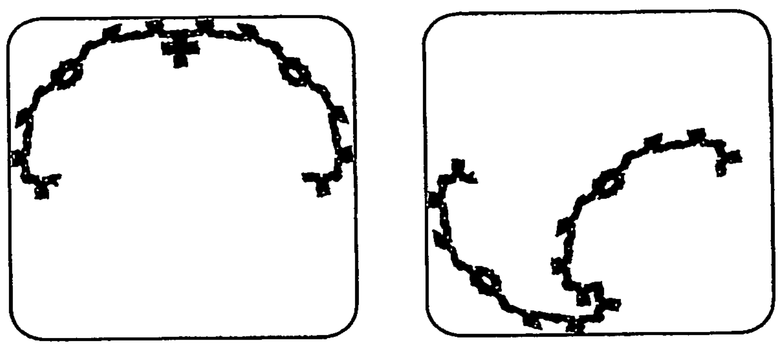


FIGURE 9. The two images after seven iterations.

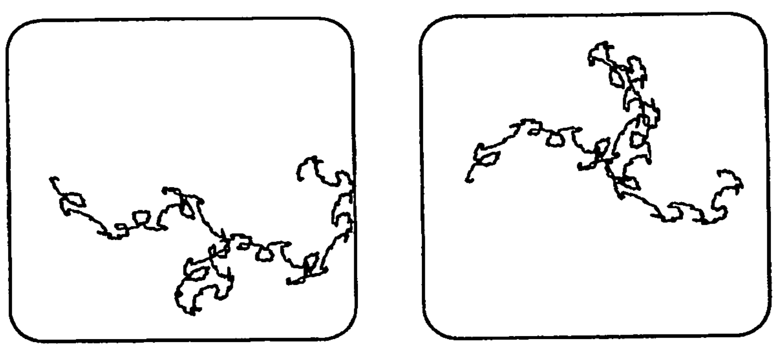


FIGURE 10. Images on the two screens $\square_1$ and $\square_2$ after a certain number $L > 20$ of iterations. Such pictures are typical of the "stationary state" at the printed resolution.

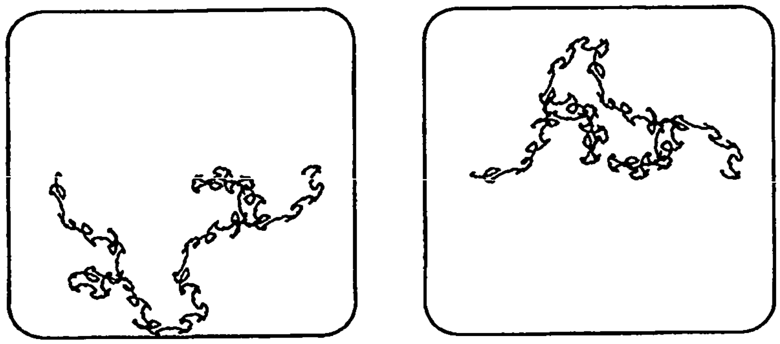


FIGURE 11. Images on the two screens after $L + 1$ iterations.

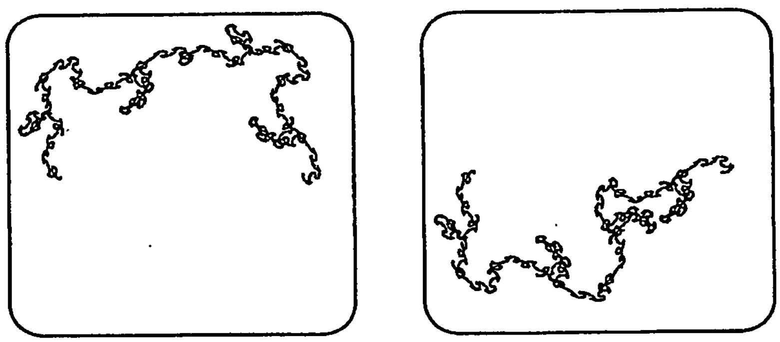


FIGURE 12. Images on the two screens after $L + 2$ iterations.

This example illustrates some typical features of the theory in this paper. (i) Keeping attention on the sequence of images that occur in one screen, we see a sequence of images, *one per iteration.* (ii) After sufficient iterations for the system to have settled into its "stationary state", each image looks like a finite resolution rendering of a (different) fractal set, each fractal belonging to the same family, in the present case a family of continuous curves.(iii) In fact, it follows from the theory that the pictures in this example correspond to curves with fractal dimension between one and two, with this property: for any $\epsilon > 0$ the curve is the union of "little" curves (i.e ones such that the distance apart of any two points is no more than ϵ) each of which is an affine transformation of one of at most two continuous closed paths in $\mathbb{R}^2$. (iv)The successive images, or rather the abstract objects they represent, are related as follows: all lie upon a single object, itself a component of an IFS attractor of a certain IFS in a certain space. This component is called a *superfractal*, the images are distributed upon it according to the invariant probability measure of the IFS, and *the images produced by the algorithm are distributed according to this measure.* (v) The images produced in the "stationary state" are

independent of the starting images. For example, if the initial images in the example had been of a dot or a line instead of a fish, and the same sequence of random choices had been made, then the images produced in Figures 10, 11, and 12 would have been the same at the printed resolution.

1.3. The structure of this paper. The main contents of this paper, while conceptually not very difficult, involves potentially elaborate notation because we deal with iterated function systems (IFSs) made of IFSs and probability measures on spaces of probability measures. So a material part of our effort has been towards a simplified notation. Thus, below, we set out some notation and conventions that we use throughout.

The core machinery that we use is basic IFS theory, as described in [5] and [12]. So in Section 2 we review relevant parts of this theory, using notation and organization that extends to and simplifies later material. To keep the structural ideas clear, we restrict attention to IFSs with strictly contractive transformations and constant probabilities. Of particular relevance to this paper, we explain what is meant by *the random iteration algorithm*.

(** Check later.) We begin Section 3 with a discussion of the code trees that underlie random fractals. We show that there exists a special class of trees, called V-variable trees, where V is an integer. Unlike general trees, these objects can be characterized in terms of an IFS acting on an appropriate space. What are these trees like? At each level they have at most V distinct subtrees! The IFS enables one to put measure attractors on the set of such trees, such that the trees can be sampled by means of the random iteration algorithm.

(**Check later.) The discussion of trees in the first few subsections of Section 3 is recapitulated several times over in the later subsections, where the same basic IFS theory is applied in more and more elaborate settings. We arrive at the formal concepts of V-variable fractals and super-fractals.

** In Section 4 we describe some generalizations and potential applications of the theory.

1.4. Some Notation. We use notation and terminology consistent with [12].

Throughout we reserve the symbols M, N, and V for positive integers. We will use the variables $m \in \{1, 2, ..., M\}$, $n \in \{1, 2, ..., N\}$, and $v \in \{1, 2, ..., V\}$.

Throughout we use an underlying metric space $(\mathbb{X}, d_{\mathbb{X}})$ where the space is $\mathbb{X}$ and the metric is $d = d_{\mathbb{X}}$. Except where otherwise stated, $(\mathbb{X}, d_{\mathbb{X}})$ is assumed to be a compact metric space. We write $\mathbb{X}^V$ to denote the compact metric space

$$\underbrace{\mathbb{X} \times \mathbb{X} \times ... \times \mathbb{X}}_{V \text{ TIMES}}. \tag{1.9}$$

with metric

$$d(x, y) = d_{\mathbb{X}^V}(x, y) = Max\ \{d_{\mathbb{X}}(x_v, y_v) : v = 1, 2, ..., V\},\ \forall x, y \in \mathbb{X}^M, \tag{1.10}$$

where $x = (x_1, x_2, ...x_V)$ and $y = (y_1, y_2, ...y_V)$.

In some applications, to computer graphics for example, $(\mathbb{X}, d_{\mathbb{X}})$ is a bounded region in $\mathbb{R}^2$ with the Euclidean metric. We will be concerned with continuous transformations $f : \mathbb{X} \rightarrow \mathbb{X}$, for example affine and projective transformations acting on $\mathbb{R}^2$.

Let $\mathbb{S} = \mathbb{S}(\mathbb{X})$ denote the set of all subsets of $\mathbb{X}$, and let $C \in \mathbb{S}$. We extend the definition of a function $f : \mathbb{X} \to \mathbb{X}$ to $f : \mathbb{S} \to \mathbb{S}$ by

$$f(C) = \{f(x) : x \in C\} \tag{1.11}$$

Let $\mathbb{H}$=$\mathbb{H}(\mathbb{X})$ denote the set of non-empty compact subsets of $\mathbb{X}$. Then if $f : \mathbb{X} \to \mathbb{X}$ is continuous we have $f : \mathbb{H} \to \mathbb{H}$. We use $d_{\mathbb{H}}$ to denote the Hausdorff distance on $\mathbb{H}$ implied by the metric $d_{\mathbb{X}}$ on $\mathbb{X}$. This is defined as follows. Let A and B be any pair of sets in $\mathbb{H}$, define the distance *from A to B* to be

$$\mathcal{D}(A, B) = Max\{Min\{d_{\mathbb{X}}(x, y) : y \in B\} : x \in A\}, \tag{1.12}$$

and define

$$d_{\mathbb{H}}(A, B) = Max\{\mathcal{D}(A, B), \mathcal{D}(B, A)\}. \tag{1.13}$$

Then $(\mathbb{H}, d_{\mathbb{H}})$ is a compact metric space. We will write $(\mathbb{H}^V, d_{\mathbb{H}^V})$ to denote the V-dimensional product space constructed from $(\mathbb{H}, d_{\mathbb{H}})$ just as $(\mathbb{X}^V, d_{\mathbb{X}^V})$ is constructed from $(\mathbb{X}, d_{\mathbb{X}})$. When we refer to continuous, Lipshitz, or strictly contractive functions acting on $\mathbb{H}^V$ we assume that the underlying metric is $d_{\mathbb{H}^V}$.

We will in a number of places start from a function acting on a space, and extend its definition to make it act on other spaces, while leaving the symbol unchanged as above.

Let $\mathbb{B} = \mathbb{B}(\mathbb{X})$ denote the set of Borel subsets of $\mathbb{X}$. Let $\mathbb{P} = \mathbb{P}(\mathbb{X})$ or $\mathbb{M} = \mathbb{M}(\mathbb{X})$ (depending on the intended interpretation as indicated below) denote the set of Borel probability measures on $\mathbb{X}$. In some applications to computer imaging one sets $\mathbb{X} = [0, 1] \times [0, 1] \subset \mathbb{R}^2$ and identifies a black and white image with a member of $\mathbb{H}(\mathbb{X})$. Greyscale images are identified with members of $\mathbb{M}(\mathbb{X})$. Probability measures on images are identified with $\mathbb{P}(\mathbb{H}(\mathbb{X}))$ or $\mathbb{P}(\mathbb{M}(\mathbb{X}))$.

Let $d_{\mathbb{P}(\mathbb{X})}$ denote the Monge-Kantorovitch metric on $\mathbb{P}(\mathbb{X})$. This is defined as follows. Let μ and ν be any pair of measures in $\mathbb{P}$. Then

(1.14)

$$d_{\mathbb{P}}(\mu, \nu) = Sup\{\int_{\mathbb{X}} f d\mu - \int_{\mathbb{X}} f d\nu : f : \mathbb{X} \to \mathbb{R}, |f(x) - f(y)| < d_{\mathbb{X}}(x, y) \ \forall\ x, y \in \mathbb{X}\}.$$

Then $(\mathbb{P}, d_{\mathbb{P}})$ is a compact metric space, as is $(\mathbb{M}, d_{\mathbb{M}})$. We define the push-forward map $f : \mathbb{P}(\mathbb{X}) \to \mathbb{P}(\mathbb{X})$ by

$$f(\mu) = \mu \circ f^{-1} \ \forall\ \mu \in \mathbb{P}(\mathbb{X}). \tag{1.15}$$

Again here we have extended the domain of action of the function $f : \mathbb{X} \to \mathbb{X}$.

We will use such spaces as $\mathbb{P}(\mathbb{H}^V)$, and $\mathbb{P}((\mathbb{M}(\mathbb{X}))^V)$ (or $\mathbb{H}(\mathbb{H}^V)$, and $\mathbb{H}(\mathbb{M}^V)$ depending on the context). These spaces may at first seem somewhat Baroque, but as we shall see, they are very natural. In each case we assume that the metric of a space is deduced from the space from which it is built, as above, down to the metric on the lowest space $\mathbb{X}$, and often we drop the subscript on the metric without ambiguity. So for example, we will write

$$d(A, B) = d_{\mathbb{H}((\mathbb{M}(\mathbb{X}))^V)}(A, B) \ \ \forall\ A, B \in \mathbb{H}((\mathbb{M}(\mathbb{X}))^V). \tag{1.16}$$

We also use the following common symbols:
$\mathbb{N} = \{1, 2, 3, ...\}$, and $\mathbb{Z} = \{... - 2, -1, 0, 1, 2, ...\}$.

2. Iterated Function Systems

2.1. Iterated Function Systems. In this section we review relevant information about IFS. To clarify the essential ideas we consider the case where all mappings are contractive, but indicate in later sections how these ideas can be generalized. The machinery and ideas introduced here are applied repeatedly later on in more elaborate settings.

Let

$$F = \{\mathbb{X}; f_1, f_2, ..., f_M; p_1, p_2, ..., p_M\} \tag{2.1}$$

denote an IFS with probabilities. The functions $f_m : \mathbb{X} \to \mathbb{X}$ are contraction mappings with fixed constant $0 \leq l < 1$; that is

$$d(f_m(x), f_m(y)) \leq l \cdot d(x,y) \ \forall x, y \in \mathbb{X}, \forall m \in \{1, 2, ..., M\}. \tag{2.2}$$

The p_m's are probabilities, with

$$\sum_{m=1}^{M} p_m = 1, \ p_m \geq 0 \ \forall m. \tag{2.3}$$

We define mappings $F : \mathbb{H}(\mathbb{X}) \to \mathbb{H}(\mathbb{X})$ and $F : \mathbb{P}(\mathbb{X}) \to \mathbb{P}(\mathbb{X})$ by

$$F(K) = \bigcup_{m=1}^{M} f_m(K) \quad \forall K \in \mathbb{H}, \tag{2.4}$$

and

$$F(\mu) = \sum_{m=1}^{M} p_m f_m(\mu) \quad \forall \mu \in \mathbb{P}. \tag{2.5}$$

In the latter case note that the weighted sum of probability measures is again a probability measure (greyscale image).

Theorem 1. [5] *The mappings $F : \mathbb{H}(\mathbb{X}) \to \mathbb{H}(\mathbb{X})$ and $F : \mathbb{P}(\mathbb{X}) \to \mathbb{P}(\mathbb{X})$ are both contractions with factor $0 \leq l < 1$. That is,*

$$d(F(K), F(L)) \leq l \cdot d(K, L) \ \forall \ K, L \in \mathbb{H}(\mathbb{X}), \tag{2.6}$$

and

$$d(F(\mu), F(\nu)) \leq l \cdot d(\mu, \nu) \ \forall \ \mu, \nu \in \mathbb{P}(\mathbb{X}). \tag{2.7}$$

As a consequence, there exists a unique nonempty compact set $A \in \mathbb{H}(\mathbb{X})$ such that

$$F(A) = A, \tag{2.8}$$

and a unique measure $\mu \in \mathbb{P}(\mathbb{X})$ such that

$$F(\mu) = \mu. \tag{2.9}$$

The support of μ is contained in, or equal to A, with equality when all of the probabilities p_m are strictly positive.

Definition 1. *The set A in Theorem 1 is called the set attractor of the IFS F, and the measure μ is called the measure attractor of F.*

We will use the term *attractor of an IFS* to mean either the set attractor or the measure attractor. We will also refer informally to the set attractor of an IFS as a *fractal* and to its measure attractor as a *fractal measure*. Furthermore, we say that the set attractor of an IFS is a *deterministic* fractal. This is in distinction to *random* fractals, and V-variable random fractals which are the main goal of this paper.

There are two main types of algorithm for the practical computation of attractors of IFS that we term *deterministic algorithms* and *random iteration algorithms*, also known as backward and forward algorithms, c.f. [15]. These terms should not be confused with the type of fractal that is computed by means of the algorithm. Both deterministic and random iteration algorithms may be used to compute deterministic fractals, and as we discuss later, a similar remark applies to our V-variable fractals.

Deterministic algorithms are based on the following:

Corollary 1. *Let $A_0 \in \mathbb{H}(\mathbb{X})$, or $\mu_0 \in \mathbb{P}(\mathbb{X})$, and define recursively*

$$A_k = F(A_{k-1}), \text{ or } \mu_k = F(\mu_{k-1}), \forall k \in \mathbb{N}, \tag{2.10}$$

respectively; then13

$$\underset{k\to\infty}{Lim} A_k = A, \text{ or } \underset{k\to\infty}{Lim} \mu_k = \mu, \tag{2.11}$$

respectively. The rate of convergence is geometrical; for example,

$$d(A_k, A) \leq l^k \cdot d(A_0, A)\ \forall k \in \mathbb{N}. \tag{2.12}$$

Example 1. *An example of a deterministic algorithm involving a simple IFS on the unit square in $\mathbb{R}^2$. See Figure 13.*

The random iteration algorithm is based on the following theorem, see [10].

Theorem 2. *Specify a starting point $x_1 \in \mathbb{X}$. Define a random orbit of the IFS to be $\{x_l\}_{l=1}^{\infty}$ where $x_{l+1} = f_m(x_l)$ with probability p_m . Then for almost all random orbits $\{x_l\}_{l=1}^{\infty}$, for any $x_1 \in \mathbb{X}$, we have:*

$$\mu(B) = \underset{l\to\infty}{Lim}\{\frac{|\{B \cap \{x_1, x_2, ..., x_l\}|}{l}\}. \tag{2.13}$$

for all $B \in \mathbb{B}(\mathbb{X})$ such that $\mu(\partial B) = 0$, where ∂B denotes the boundary of ∂B.

Remark 1. *This is equivalent by standard arguments to the following: for almost all random orbits the sequence of point measures $\frac{1}{l}(\delta_{x_1} + \delta_{x_2} + ... + \delta_{x_l})$ converges to μ, see for example* [13], *pages 11 and 12.*

Example 2. *An example of the random iteration algorithm involving a simple IFS on the unit square in $\mathbb{R}^2$. See Figure 14*

The following theorem expresses the ergodicity of the IFS F. The proof depends centrally on the uniqueness of the measure attractor, and it applies more generally, as stated for example in [11] (??) where a probabilistic proof is given.

Theorem 3. *Suppose that μ is the measure attractor for the IFS F and $B \in \mathbb{B}(\mathbb{X})$ is such that $f_m(B) \subset B\ \ \forall m \in \{1, 2, ..., M\}$. Then $\mu(B) = 0$ or 1.*

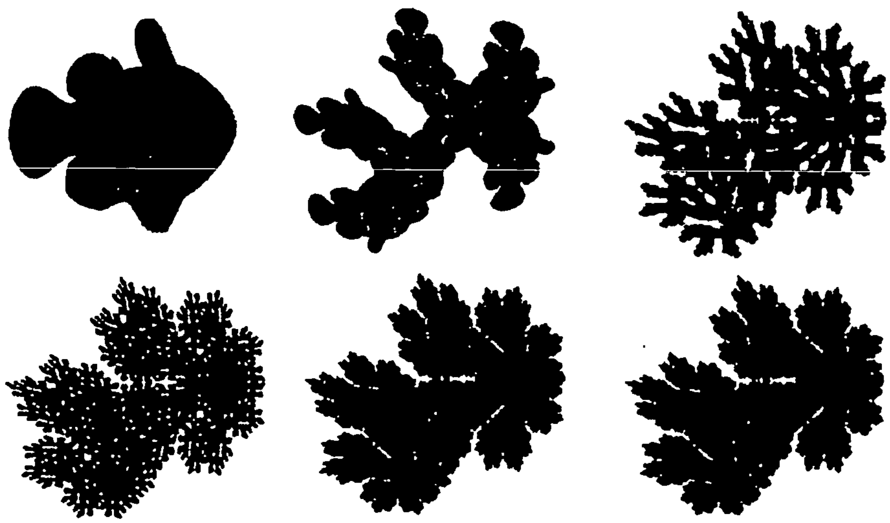


FIGURE 13. An illustration of the deterministic algorithm, also showing the texture effect.

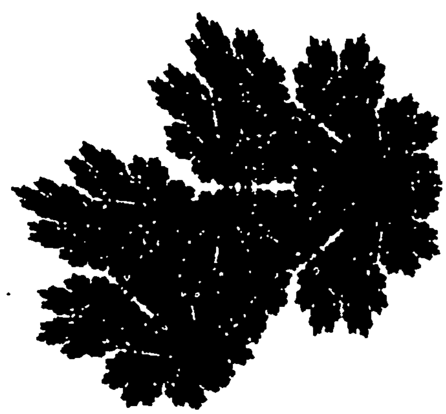


FIGURE 14. "Picture" of the measure attractor of an IFS with probabilities produced by the random iteration algorithm. The IFS is $\{\square; f_1^1, f_2^1, f_2^2; 0.4, 0.3, 0.4\}$ where the transformations are those defined in the Introduction. The measure is depicted in shades of green, from 0 (black) to 255 (bright green).

Proof. Let us define the measure $\mu\lfloor B$ (μ restricted to B) by $(\mu\lfloor B)(E) = \mu(B \cap E)$. The main point of the proof is to show that $\mu\lfloor B$ is invariant under the IFS F. (A similar result applies to $\mu\lfloor B^C$ where B^C denotes the complement of B.

If $E \subset B^C$, for any m, since $f_m(B) \subset B$,

$$f_m(\mu\lfloor B)(E) = \mu(B \cap f_m^{-1}(E)) = \mu(\emptyset) = 0. \tag{2.14}$$

Moreover, for any m,

$$\mu(B) = f_m(\mu\lfloor B)(\mathbb{X}) = f_m(\mu\lfloor B)(B). \tag{2.15}$$

It follows that

$$\begin{aligned} \mu(B) &= \sum_{m=1}^{M} p_m f_m \mu(B) \\ &= \sum_{m=1}^{M} p_m f_m(\mu\lfloor B)(B) + \sum_{m=1}^{M} p_m f_m(\mu\lfloor B^C)(B) \\ &= \mu(B) + \sum_{m=1}^{M} p_m f_m(\mu\lfloor B^C)(B) \text{ (from (2.15))}. \end{aligned} \tag{2.16}$$

Hence

$$\sum_{m=1}^{M} p_m f_m(\mu\lfloor B^C)(B) = 0. \tag{2.17}$$

Hence for any measureable set $E \subset \mathbb{X}$

$$\begin{aligned} (\mu\lfloor B)(E) &= \mu(B \cap E) \\ &= \sum_{m=1}^{M} p_m f_m \mu(B \cap E) \\ &= \sum_{m=1}^{M} p_m f_m(\mu\lfloor B)(B \cap E) + \sum_{m=1}^{M} p_m f_m(\mu\lfloor B^C)(B \cap E) \\ &= \sum_{m=1}^{M} p_m f_m(\mu\lfloor B)(E) + 0 \quad \text{(using (2.16))}. \end{aligned} \tag{2.18}$$

Thus $\mu\lfloor B$ is invariant and so is either the zero measure or for some constant $c \geq 1$ we have $c\mu\lfloor B = \mu$ (by uniqueness)$= \mu\lfloor B + \mu\lfloor B^C$. This implies $\mu\lfloor B^C = 0$ and in particular $\mu(B^C) = 0$ and $\mu(B) = 1$. □

Theorem 3 provides a simple explanation for an computer graphics effect that may be seen when the set attractor of an IFS on $\mathbb{R}^2$ is computed using a deterministic algorithm, as illustrated in the following example. A similar effect can also occur in the case of V-variable fractals.

Example 3. *Example illustrating the texture effect.*

**An important number concerning fractals is the fractal dimension. Not only is it an important invariant, but in practice says something about the look and feel of an attractor.

Definition 2. ***Fractal Dimension*

Definition 3. ** *Open set condition*

Theorem 4. ***Fractal Dimension*

2.2. **Code Space.** A good way of looking at an IFS F as in (2.1) is in terms of the associated *code space* $\Sigma = \{1, 2, ..., M\}^{\infty}$. Members of Σ are infinite sequences from the alphabet $\{1, 2, ..., M\}$ and indexed by $\mathbb{N}$. We equip Σ with the metric d_{Σ} defined for $\omega \neq \varkappa$ by

$$d_{\Sigma}(\omega, \varkappa) = \frac{1}{2^k}, \tag{2.19}$$

where k is the index of the first symbol at which $\omega \neq \varkappa$ differ. Then (Σ, d_{Σ}) is a compact metric space.

Theorem 5. *Let A denote the set attractor of the IFS F. Then there exists a continuous onto mapping $F : \Sigma \to A$, defined for all $\sigma_1\sigma_2\sigma_3... \in \Sigma$ by*

$$F(\sigma_1\sigma_2\sigma_3...) = \lim_{k\to\infty} f_{\sigma_1} \circ f_{\sigma_2} \circ ... \circ f_{\sigma_k}(x). \tag{2.20}$$

The limit is independent of $x \in \mathbb{X}$ and the convergence is uniform in x.

Definition 4. *The point $\sigma_1\sigma_2\sigma_3... \in \Sigma$ is called an address of the point $F(\sigma_1\sigma_2\sigma_3...) \in A$.*

Note that $F : \Sigma \to A$ is not in general one-to-one.

The following theorem characterizes the measure attractor of the IFS F as the push-foward, under $F : \Sigma \to A$, of an elementary measure $\rho \in \mathbb{P}(\Sigma)$, the measure attractor of a fundamental IFS on Σ.

Theorem 6. *For each $m \in \{1, 2, ..., M\}$ define the shift operator $s_m : \Sigma \to \Sigma$ by*

$$s_m(\sigma_1\sigma_2\sigma_3...) = m\sigma_1\sigma_2\sigma_3... \tag{2.21}$$

$\forall\sigma_1\sigma_2\sigma_3... \in \Sigma$. Then s_m is a contraction mapping with contractivity factor $\frac{1}{2}$. Consequently

$$S := \{\Sigma; s_1, s_2, ..., s_M; p_1, p_2, ..., p_M\} \tag{2.22}$$

is an IFS. Its set attractor is Σ. Its measure attractor is the unique measure $\rho \in \mathbb{P}(\Sigma)$ such that

$$\begin{aligned} \rho(\{\omega_1\omega_2\omega_3... &\in \Sigma | \omega_1 = \sigma_1, \omega_2 = \sigma_2, ..., \omega_k = \sigma_k\}) \\ &= p_{\sigma_1} \cdot p_{\sigma_2} \cdot ... \cdot p_{\sigma_k} \end{aligned} \tag{2.23}$$

$\forall k \in \mathbb{N}$, $\forall\sigma_1, \sigma_1, ..., \sigma_k \in \{1, 2, ..., M\}$.

If μ is the measure attractor of the IFS F, with $F : \Sigma \to A$ defined as in Theorem 5, then

$$\mu = F(\rho). \tag{2.24}$$

We call S the *shift* IFS on code space. It has been well studied in the context of information theory and dynamical systems, see for example [14], and results can often be lifted to the IFS F. **Example.

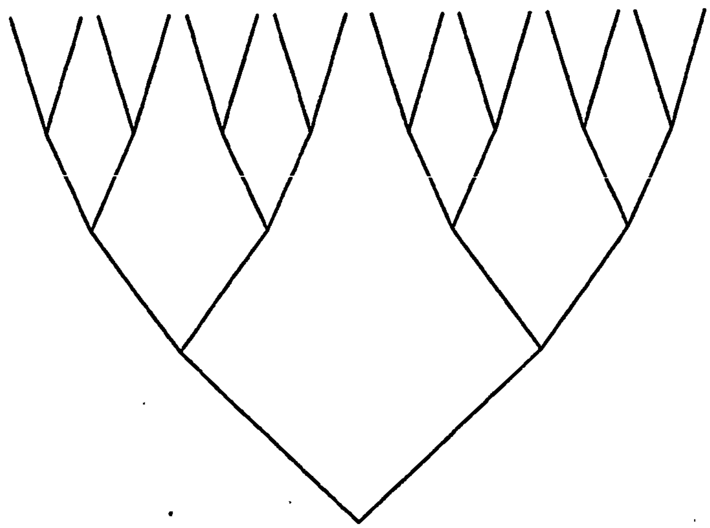


FIGURE 15. Illustration of a 2-variable tree, up to level 5.

3. TREES OF ITERATED FUNCTION SYSTEMS AND SUPERFRACTALS

3.1. Trees of IFSs. Let (X, d_X) be a compact metric space, and let M and N be a positive integers. For $n \in \{1, 2, ..., N\}$ let F^n denote the IFS

$$\{\mathbb{X}; f_1^n, f_2^n, ...f_M^n; p_1^n, p_2^n, ...p_M^n\} \tag{3.1}$$

where each $f_m^n : \mathbb{X} \to \mathbb{X}$ is a Lipshitz function with Lipshitz constant $0 \leq l < 1$ and the p_m^n 's are probabilities with

$$\sum_{m=1}^{M} p_m^n = 1,\ p_m^n \geq 0\ \forall\ m, n. \tag{3.2}$$

Let

$$\mathcal{F} = \{\mathbb{X}; F^1, F^2, ..., F^N; P^1, P^2, ..., P^N\}, \tag{3.3}$$

where the P^n's are probabilitiies with

$$\sum_{n=1}^{N} P^n = 1,\ P^n \geq 0\ \ \forall n. \tag{3.4}$$

$P^{(n)} > 0\ \forall n \in \{1, 2, ..., N\}$ and $\sum_{n=1}^{N} P^{(n)} = 1$. We call $\mathcal{F}$ a *superIFS*, for reasons that will become clear as we proceed. This is *not* a standard IFS because the constituent "functions", the F^n's, themselves standard IFS's, are to be composed in a special way, to be explained.

We associate with $\mathcal{F}$ the following collection of labelled trees Ω. Let $T = T_M$ denote the (M-fold) tree of finite sequences from $\{1, 2, ..., M\}$, including the empty sequence $\emptyset$. A 2-fold tree is illustrated in Figure 15. For $\sigma = \sigma_1\sigma_2...\sigma_k \in T$

let $|\sigma| = k$. If also $\tau = \tau_1\tau_2...\tau_l \in T$ then $\sigma\tau$ is the concatenated sequence $\sigma_1\sigma_2...\sigma_k\tau_1\tau_2...\tau_l$. A *code tree* is a map $\sigma : T \to \{1, 2, ..., N\}$ and we write

$$\Omega = \{\sigma | \sigma : T \to \{1, 2, ..., N\}\} \tag{3.5}$$

for the set of all such code trees.

We define a metric on Ω by, for $\omega \neq \varkappa$,

$$d_\Omega(\omega, \varkappa) = \frac{1}{2^k} \tag{3.6}$$

where k is the index of the first symbol at which the strings that represent ω and $\varkappa$ disagree. Then (Ω, d_Ω) is a compact metric space.

A *construction tree* of $\mathcal{F}$ is a code tree wherein the symbols $1, 2, ...,$ and N are replaced by the respective IFSs $F^1, F^2, ...,$ and F^N. A construction tree consists of nodes and branches, where each node is labelled by one of the IFSs belonging to $\mathcal{F}$. We will sometimes associate the M branches that lie below a node with the constituent functions of the IFS of that node; taken in order.

We use the notation $\mathcal{F}(\Omega)$ to denote the set of construction trees of $\mathcal{F}$. For $\sigma \in \Omega$ we write $\mathcal{F}(\sigma)$ to denote the corresponding construction tree. We will use the *same notation* $\mathcal{F}(\sigma)$ to denote the random fractal set associated with the construction tree $\mathcal{F}(\sigma)$, as described in the next section.

3.2. **Random Fractals.** In this section we describe the canonical random fractal sets and measures associated with the superIFS $\mathcal{F}$ in (3.3). These sets and measures are defined using random code trees obtained as follows. We put a probability measure $\rho \in P(\Omega)$ on the set of code trees, by setting $\sigma(\tau) = n$ with probability P^n, for $n \in \{1, 2, ..., N\}$ independently at each node of the tree $\tau. \in T$. Thus we are able to speak of *the random code trees Ω with probability distribution ρ*, and of *selecting trees $\sigma \in \Omega$ according to ρ.*

Let a superIFS $\mathcal{F}$ be given as in (3.3), let $k \in \mathbb{N}$, and define

$$\mathcal{F}_k : \Omega \times \mathbb{H}(\mathbb{X}) \to \mathbb{H}(\mathbb{X}), \tag{3.7}$$

by

$$\mathcal{F}_k(\sigma)(K) = \bigcup_{\{\tau \in T_M | \, |\tau| = k\}} f_{\tau_1}^{\sigma(\emptyset)} \circ f_{\tau_2}^{\sigma(\tau_1)} \circ ... \circ f_{\tau_k}^{\sigma(\tau_1\tau_2...\tau_{k-1})}(K) \tag{3.8}$$

$\forall\, \sigma \in \Omega$ and $K \in \mathbb{H}(\mathbb{X})$. (The set $\mathcal{F}_k(\sigma)(K)$ is obtained by taking the union of the compositions of the functions occuring on the branches of the construction tree $\mathcal{F}(\sigma)$ starting at the top and working down to the k^{th} level, acting upon K.)

In a similar way, with measures in place of sets, and unions of sets replaced by sums of measures weighed by probabilities, we define

$$\tilde{\mathcal{F}}_k : \Omega \times \mathbb{P}(\mathbb{X}) \to \mathbb{P}(\mathbb{X}) \tag{3.9}$$

by

$$\tilde{\mathcal{F}}_k(\sigma)(\eta) = \sum_{\{\tau \in T_M | \, |\tau| = k\}} \tilde{f}_{\tau_1}^{\sigma(\emptyset)} \circ \tilde{f}_{\tau_2}^{\sigma(\tau_1)} \circ ... \circ \tilde{f}_{\tau_k}^{\sigma(\tau_1\tau_2...\tau_{k-1})}(\eta) \tag{3.10}$$

$\forall\, \sigma \in \Omega$ and $\eta \in \mathbb{P}(\mathbb{X})$ where

$$\tilde{f}_m^n = p_m^n \cdot f_m^n \tag{3.11}$$

$\forall\, m \in \{1, 2, ..., M\}$ and $n \in \{1, 2, ..., N\}$. Note that the $\tilde{\mathcal{F}}_k(\sigma)(\eta)$ all have unit mass because of the p_m^n in Equation 3.11.

Theorem 7. *Let sequences of functions$\{\mathcal{F}_k\}$and $\{\tilde{\mathcal{F}}_k\}$ be defined as above. Then both the limits*

$$\mathcal{F}(\sigma) = \underset{k\to\infty}{Lim}\{\mathcal{F}_k(\sigma)(K)\}, \text{ and } \tilde{\mathcal{F}}(\sigma) = \underset{k\to\infty}{Lim}\{\tilde{\mathcal{F}}_k(\sigma)(\eta)\}, \tag{3.12}$$

exist, independently of K and η, and the convergence (in the Hausdorff and Monge-Kantorovitch metrics, respectively,)is uniform in σ, K, and η,. The resulting functions

$$\mathcal{F} : \Omega \to \mathbb{H}(\mathbb{X}) \text{ and } \tilde{\mathcal{F}} : \Omega \to \mathbb{P}(\mathbb{X}) \tag{3.13}$$

are continuous.

Proof. Make repeated use of the fact that, for fixed $\sigma \in \Omega$, both mappings are compositions of contraction mappings of contractivity l, by Theorem 1. □

Let

$$\mathfrak{H} = \{\mathcal{F}(\sigma) \in \mathbb{H}(\mathbb{X}) | \sigma \in \Omega\}, \text{ and } \tilde{\mathfrak{H}} = \{\tilde{\mathcal{F}}(\sigma) \in \mathbb{P}(\mathbb{X}) | \sigma \in \Omega\}. \tag{3.14}$$

Similarly let

$$\mathfrak{P} = \mathcal{F}(\rho) = \rho \circ \mathcal{F}^{-1} \in \mathbb{P}(\mathfrak{H}), \text{ and } \tilde{\mathfrak{P}} = \tilde{\mathcal{F}}(\rho) = \rho \circ \tilde{\mathcal{F}}^{-1} \in \mathbb{P}(\tilde{\mathfrak{H}}). \tag{3.15}$$

Definition 5. *The sets $\mathfrak{H}$ and $\tilde{\mathfrak{H}}$ are called the sets of random fractal sets and random fractal measures, respectively, associated with the superIFS $\mathcal{F}$. These random fractal sets and measures are said to be distributed according to $\mathfrak{P}$ and $\tilde{\mathfrak{P}}$, respectively.*

Random fractal sets and measures are hard to compute. There does not appear to be a general simple forwards (random iteration) algorithm for practical computation of approximations to them in two-dimensions with affine maps, for example. The reason for this difficulty lies with the inconvenient manner in which the shift operator acts on trees $\sigma \in \Omega$ relative to the expressions (3.8) and (3.10).

Theorem 8. *** Fractal dimension of random fractals, say in the case of similitudes.*

3.3. Contraction Mappings on Code Trees and the Space Ω_V. Let $V \in \mathbb{N}$. This parameter will describe the *variability* of the trees and fractals that we are going to introduce. Let

$$\Omega^V = \underbrace{\Omega \times \Omega \times ... \times \Omega}_{V \text{ TIMES}}. \tag{3.16}$$

We refer to an element of Ω^V as a *grove*. In this section we describe a certain IFS on Ω^V, and discuss its set attractor Ω_V: its points are (V-tuples of) code trees that we will call V*-groves*.

One reason that we are interested in Ω_V is that, as we shall see later, the set of trees that occur in its components, called V*-trees*, provides the appropriate code space for a V-variable superfactal.

We are going to use here and in later sections an index set

$$\mathcal{A} := \{1, 2, ..., N\}^V \times \{1, 2, ..., V\}^{MV}. \tag{3.17}$$

A typical index $a \in \mathcal{A}$ will be denoted

$$a = (n_1, n_2, ..., n_V) \times (v_1, v_2, ..., v_{MV}) \tag{3.18}$$

where each $n_v \in \{1, 2, ..., N\}$ and each $v_j \in \{1, 2, ..., V\}$.

We are also going to need a set of probabilities $\{P^a | a \in \mathcal{A}\}$, with

$$\sum_{a \in \mathcal{A}} P^a = 1,\ P^a \geq 0\ \forall a \in \mathcal{A}. \tag{3.19}$$

These probabilities may be more or less complicated. Some of our results are specifically restricted to the case

$$P^a = \frac{P^{n_1} P^{n_2} ... P^{n_V}}{V^{MV}}, \tag{3.20}$$

which uses only the set of probabilities $\{P^{(1)}, P^{(2)}, ..., P^{(N)}\}$ belonging to the superIFS (3.3).

Theorem 9. *For each $n \in \{1, 2, ..., N\}$ define the n^{th} shift mapping $\xi_n : \Omega^M \to \Omega$ by*

$$(\xi_n(\omega))(\varnothing) = n \text{ and } (\xi_n(\omega))(m\tau) = \omega_m(\tau)\ \forall\ \tau \in T, m \in \{1, 2, ..., M\}, \tag{3.21}$$

for $\omega = (\omega_1, \omega_2, ..., \omega_M) \in \Omega^M$. That is, the mapping ξ_n creates a code tree with its top node labelled n attached directly to the M trees $\omega_1, \omega_2, ..., \omega_M$. For each $a \in \mathcal{A}$ define $\eta_a := \Omega^V \to \Omega^V$ by

$$\begin{aligned} \eta_a(\omega_1, \omega_2, ..., \omega_V) \ = \ & (\xi_{n_1}(\omega_{v_1}, \omega_{v_2}, ..., \omega_{v_M}), \\ & \xi_{n_2}(\omega_{v_{M+1}}, \omega_{v_{M+2}}, ..., \omega_{v_{2M}}), ... \\ & \xi_{n_V}(\omega_{v_{MV-V+1}}, \omega_{v_{MV-V+2}}, ..., \omega_{v_{MV}})). \end{aligned} \tag{3.22}$$

Then $\eta_a : \Omega^V \to \Omega^V$ is a contraction mapping with Lipshitz constant $\frac{1}{2}$and consequently

$$\Phi := \{\Omega^V; \eta_a, P^a, a \in \mathcal{A}\} \tag{3.23}$$

is an IFS of strictly contractive maps.

Let the set attractor and the measure attractor of the IFS Φ be denoted by Ω_V, and μ_V respectively. Clearly, $\Omega_V \in \mathbb{H}(\Omega^V)$ while $\mu_V \in \mathbb{P}(\Omega^V)$. The elements of Ω_V are certain V-tuples of M-fold code trees on an alphabet of N symbols, which we characterize in Theorem 12. But we think of them as groves of trees.

Definition 6. *Ω_V is called the space of V-groves. An element of Ω_V is called a V-grove. An element of a component of a V-grove is called a V-tree.*

For all $v \in \{1, 2, ..., V\}$, let us define $\Omega_{V,v} \subset \Omega_V$ to be the set of v^{th} components of groves in Ω_V.

Theorem 10. *For all $v \in \{1, 2, ..., V\}$ we have*

$$\Omega_{V,v} = \Omega_{V,1} := \{\text{set of all V-trees}\}. \tag{3.24}$$

When the probabilities $\{P^a | a \in \mathcal{A}\}$ obey (3.20), the distribution of trees $\omega \in \Omega$ that occur in the v^{th} components of groves produced by the random iteration algorithm corresponding to the IFS Φ converge weakly to the marginal probability measure

$$\mu_{V,1}(B) := \mu_V(B, \Omega, \Omega, ..., \Omega) \forall B \in \mathbb{B}(\Omega), \tag{3.25}$$

independently of v.

Example 4. *Describe how this works in practice. Give an example, with one code twice as probable as the other.*

Theorem 11. *The set of V-trees converges to the set of all trees Ω in the metric of $\mathbb{H}(\Omega)$, that is*

$$\underset{V\to\infty}{Lim}\, \Omega_{V,1} = \Omega. \tag{3.26}$$

Moreover, when the probabilities $\{P^a | a \in \mathcal{A}\}$ obey (3.20),

$$\underset{V\to\infty}{Lim}\, \mu_{V,1} = \rho, \tag{3.27}$$

where ρ is the stationary measure on trees introduced in Section 3.2, and convergence is in the metric of $\mathbb{H}(\Omega)$.

Proof. Let $V \geq M^k$. Then both

$$d_{\mathbb{H}(\Omega)}(\Omega_{V,1}, \Omega) \leq \frac{1}{2^V}, \tag{3.28}$$

and

$$d_{\mathbb{P}(\Omega)}(\mu_{V,1}, \rho) \leq \frac{1}{2^V}. \tag{3.29}$$

□

Theorem 12. *Let $\sigma \in \Omega^V$. Then $\sigma \in \Omega_V$ if and only if the number of distinct subtrees of the trees in the V-grove σ, at any fixed depth, is at most V. Given any tree $\tilde{\sigma} \in \Omega$ with the property that, at any depth, it contains at most V distinct subtrees, then there is a V-grove $\sigma \in \Omega_V$ whose first component is $\tilde{\sigma}$.*

Let $\Sigma_V = \mathcal{A}^\infty$. This is simply the code space corresponding to the IFS Φ. From Theorem 5 there exists a continuous onto mapping $\Phi : \Sigma_V \to \Omega_V$ defined by

$$\Phi(a_1 a_2 a_3 ...) = \underset{k\to\infty}{Lim} \eta_{a_1} \circ \eta_{a_2} \circ ... \circ \eta_{a_k}(\omega) \tag{3.30}$$

for all $a_1 a_2 a_3 ... \in \Sigma_V$, for any $\omega \in \Omega$.

In the terminology of section 2.2 the sequence $a_1 a_2 a_3 .. \in \Sigma_V$ is an address of the V-grove $\Phi(a_1 a_2 a_3 ...) \in \Omega_V$ and Σ_V is the code space for the set of V-groves Ω_V. However, as an *address* space for indexing the V-groves of Ω_V it is not particularly helpful because in general $\Phi : \Sigma_V \to \Omega_V$ is not one-to-one as various compositions of its functions can coincide, as the following example shows.

Example 5. *Example showing that $\Phi : \Sigma_V \to \Omega_V$ is not 1-1 in general.*

3.4. Contraction Mappings on $\mathbb{H}^V$ and the Superfractal Set $\mathfrak{H}_{V,1}$.

Theorem 13. *Let $V \in \mathbb{N}$, let $\mathcal{A}$ be the index set introduced in (3.17), let $\mathcal{F}$ be given as in (3.3), and let probabilities $\{P^a | a \in \mathcal{A}\}$ be given as in (3.19). Define*

$$\mathcal{F}^a : \mathbb{H}^V \to \mathbb{H}^V \tag{3.31}$$

by

$$\mathcal{F}^a(K) = (\bigcup_{m=1}^{M} f_m^{n_1}(K_{v_m}), \bigcup_{m=1}^{M} f_m^{n_2}(K_{v_{M+m}}), \dots \bigcup_{m=1}^{M} f_m^{n_V}(K_{v_{VM-M+m}})) \tag{3.32}$$

$\forall K = (K_1, K_2, ..., K_V) \in \mathbb{H}^V, \forall a \in \mathcal{A}$. Then

$$\mathcal{F}_V = \{\mathbb{H}^V; \mathcal{F}^a, P^a, a \in \mathcal{A}\} \tag{3.33}$$

is an IFS with contractivity factor l.

Proof. We begin by noting that, $\forall (K_1, K_2, ..., K_M), (L_1, L_2, ..., L_M) \in \mathbb{H}^M$,

$$\begin{aligned} & d_{\mathbb{H}}(\bigcup_{m=1}^{M} f_m^n(K_m), \bigcup_{m=1}^{M} f_m^n(L_m)) \\ \le\ & \underset{m}{Max}\{d_{\mathbb{H}}(f_m^n(K_m), f_m^n(L_m))\} \\ \le\ & \underset{m}{Max}\{l \cdot d_{\mathbb{H}}(K_m, L_m)\} \\ =\ & l \cdot d_{\mathbb{H}^M}(K, L). \end{aligned} \tag{3.34}$$

Hence, $\forall (K_1, K_2, ..., K_V), (L_1, L_2, ..., L_V) \in \mathbb{H}^V$,

$$\begin{aligned} & d_{\mathbb{H}^V}(\mathcal{F}^a(K_1, K_2, ..., K_V), \mathcal{F}^a(L_1, L_2, ..., L_V)) \\ =\ & \underset{v}{Max}\{d_{\mathbb{H}}(\bigcup_{m=1}^{M} f_m^{n_v}(K_{v_{vM-M+m}}), \bigcup_{m=1}^{M} f_m^{n_v}(L_{v_{vM-M+m}}))\} \\ \le\ & \underset{v}{Max}\{l \cdot d_{\mathbb{H}^M}((K_{v_{vM-M+1}}, K_{v_{vM-M+2}}, ..., K_{v_{vM}}), \\ & (L_{v_{vM-M+1}}, L_{v_{vM-M+2}}, ..., L_{v_{vM}}))\} \\ \le\ & l \cdot d_{\mathbb{H}^V}((K_1, K_2, ..., K_V), (L_1, L_2, ..., L_V)). \end{aligned} \tag{3.35}$$

□

The theory of IFS in section 2.1 applies to the IFS $\mathcal{F}_V$. It possesses a unique set attractor $\mathfrak{H}_V \in \mathbb{H}(\mathbb{H}^V)$, and a unique measure attractor $\mathfrak{P}_V \in \mathbb{P}(\mathbb{H}^V)$. The random iteration algorithm corresponding to the IFS $\mathcal{F}_V$ may be used to approximate sequences of points (vectors of compact sets) in $\mathfrak{H}_V$ distributed according to the probability measure $\mathfrak{P}_V$.

However, the individual components of these vectors in $\mathfrak{H}_V$, certain special subsets of $\mathbb{X}$, are the objects we are interested in. Accordingly, for all $v \in \{1, 2, ..., V\}$, let us define $\mathfrak{H}_{V,v} \subset \mathbb{H}$ to be the set of v^{th} components of points in $\mathfrak{H}_V$.

Theorem 14. *For all $v \in \{1, 2, ..., V\}$ we have*

$$\mathfrak{H}_{V,v} = \mathfrak{H}_{V,1}. \tag{3.36}$$

When the probabilities in the superIFS $\mathcal{F}_V$ are given by (3.20), the sets $K \in \mathbb{H}$ that occur in the v^{th} component of vectors produced by the random iteration algorithm converge weakly to the marginal probability measure

$$\mathfrak{P}_{V,1}(B) := \mathfrak{P}_V(B, \mathbb{H}, \mathbb{H}, ..., \mathbb{H}) \forall B \in \mathbb{B}(\mathbb{H}), \tag{3.37}$$

independently of v.

Definition 7. *We call $\mathfrak{H}_{V,1}$ a superfractal set. Points in $\mathfrak{H}_{V,1}$ are called V-variable fractal sets.*

Example 6. *An example that will be referred to below. Points, each one a fractal set, appear to dance around on the superfractal.*

By Theorem 5 there is a continuous mapping $\mathcal{F}_V : \Sigma_V \rightarrow \mathfrak{H}_V$ that assigns to each address in the code space Σ_V a V-tuple of compact sets in $\mathfrak{H}_V$. But this mappings is not helpful for characterizing $\mathfrak{H}_V$ because $\mathcal{F}_V : \Sigma_V \rightarrow \mathfrak{H}_V$ is not in general one-to-one, for the same reason that $\Phi : \Sigma_V \rightarrow \Omega_V$ is not one-to-one, as demonstrated in Example 5. Also Σ_V does not exhibit in any direct manner the tree-like structures that underlie V-variable fractals. The following is closer to the point. It tells us that the set of V-trees is a useful code space for V-variable fractals.

Theorem 15. *Let $\Omega_{V,v}$ denote the set of v^{th} components of members of Ω_V. Let $\mathcal{F} : \Omega \rightarrow \mathbb{H}(\mathbb{X})$ be the mapping introduced in Theorem 7. Then*

$$\mathcal{F}(\Omega_{V,1}) = \mathfrak{H}_{V,1}. \tag{3.38}$$

When the probabilities in the superIFS $\mathcal{F}_V$ are given by (3.20) we have

$$\mathcal{F}(\mu_{V,1}) = \mathfrak{P}_{V,1} \tag{3.39}$$

where $\mu_{V,1}$ is the marginal probability distribution given by (3.25).

Definition 8. *The code tree $\Phi(a_1a_2a_3...)$ is called a tree address of the V-variable fractal $\mathcal{F}(a_1a_2a_3...)$.*

The mapping $\mathcal{F} : \Omega_{V,1} \rightarrow \mathfrak{H}_{V,1}$ together with Theorem 12 allows us to characterize V-variable fractals as follows:

Theorem 16. *Any V-variable fractal is made of at most V "forms" or "shapes" at each level.*

Example 7. *Example of the Theorem, in the form of a black and white picture, pointing out the key fact!*

Theorem 17. *The set of V-variable fractal sets associated with the superIFS $\mathcal{F}$ converges to the set of random fractal sets introduced in Section 3.2; that is, in the metric of $\mathbb{H}(\mathbb{H}(\mathbb{X}))$,*

$$\lim_{V\rightarrow\infty} \mathfrak{H}_{V,1} = \mathfrak{H}. \tag{3.40}$$

Moreover, when the probabilities $\{P^a | a \in \mathcal{A}\}$ obey (3.20), then in the metric of $\mathbb{P}(\mathbb{H}(\mathbb{X}))$

$$\lim_{V\rightarrow\infty} \mathfrak{P}_{V,1} = \mathfrak{P}, \tag{3.41}$$

where $\mathfrak{P}$ is the stationary measure on random fractal sets associated with the superIFS $\mathcal{F}$

Proof. This follows at once from Theorems 7 and 11. □

Theorem 18. *Result about fractal dimension, say in the case of open set condition.*

3.5. Contraction Mappings on $\mathbb{H}^V$ and the Superfractal Measures $\tilde{\mathfrak{H}}_{V,1}$. Let $\mathbb{M}^V = (\mathbb{M}(\mathbb{X}))^V (= (\mathbb{P}(\mathbb{X}))^V)$. In this section we follow the same lines as in Section 3.4, constructing an IFS using the individual IFSs of the superIFS $\mathcal{F}$, except that here the underlying space is $\mathbb{M}^V$ instead of $\mathbb{H}^V$.

Theorem 19. *Let $V \in \mathbb{N}$, let $\mathcal{A}$ be the index set introduced in (3.17), let $\mathcal{F}$ be given as in (3.3), and let probabilities $\{P^a | a \in \mathcal{A}\}$ be given as in (3.19). Define*

$$\mathcal{F}^a : \mathbb{M}^V \to \mathbb{M}^V \tag{3.42}$$

by

(3.43)
$$\mathcal{F}^a(\mu) = (\sum_{m=1}^{M} p_m^{n_1} f_m^{n_1}(\mu_{v_m}), \sum_{m=1}^{M} p_m^{n_2} f_m^{n_2}(\mu_{v_{M+m}}), ..., \sum_{m=1}^{M} p_m^{n_V} f_m^{n_V}(\mu_{v_{MV-V+m}}))$$

$\mu = (\mu_1, \mu_2, ..., \mu_V) \in \mathbb{M}^V$. Then $\mathcal{F}^a : \mathbb{M}^V \to \mathbb{M}^V$ is a contraction mapping and

$$\tilde{\mathcal{F}}_V = \{\mathbb{M}^V; \mathcal{F}^a, P^a, a \in \mathcal{A}\} \tag{3.44}$$

is an IFS with contractivity factor l.

The set attractor of the IFS $\tilde{\mathcal{F}}_V$ is $\tilde{\mathfrak{H}}_V \in \mathbb{H}(\mathbb{M}^V)$, a subset of $\mathbb{M}^V$, a set of V-tuples of probability measures on $\mathbb{X}$. As we will see, each of these measures is supported on a V-variable fractal set belonging to the superfactal $\tilde{\mathfrak{H}}_{V,1}$.The measure attractor of the IFS $\tilde{\mathcal{F}}_V$ is a probability measure $\tilde{\mathfrak{P}}_V \in \mathbb{P}(\mathbb{M}^V)$, namely a probability measure on a set of V-tuples of normalized measures, each one a random fractal measure. The random iteration algorithm corresponding to the IFS $\tilde{\mathcal{F}}_V$ may be used to approximate sequences of points in $\tilde{\mathfrak{H}}_V$, namely vectors of measures on $\mathbb{X}$, distributed according to the probability measure $\tilde{\mathfrak{P}}_V$.

As in Section 3.4, we define $\tilde{\mathfrak{H}}_{V,v}$ to be the set of v^{th} components of sets in $\tilde{\mathfrak{H}}_V$, for $v \in \{1, 2, ..., V\}$.

Theorem 20. *For all $v \in \{1, 2, ..., V\}$ we have*

$$\tilde{\mathfrak{H}}_{V,v} = \tilde{\mathfrak{H}}_{V,1}. \tag{3.45}$$

When the probabilities in the IFS $\tilde{\mathcal{F}}_V$ are given by (3.20), the probability measures $\mu \in \mathbb{M}(\mathbb{X})$ that occur in the v^{th} component of points produced by the random iteration algorithm converge weakly to the marginal probability measure

$$\tilde{\mathfrak{P}}_{V,1}(B) := \tilde{\mathfrak{P}}_V(B, \mathbb{M}, \mathbb{M}, ..., \mathbb{M}) \forall B \in \mathbb{B}(\mathbb{M}), \tag{3.46}$$

independently of v.

Definition 9. *We call $\tilde{\mathfrak{H}}_{V,1}$ a superfractal set of measures (of variability V). Points in $\tilde{\mathfrak{H}}_{V,1}$ are called V-variable fractal measures.*

Example 8. *An example that will be referred to below. Points, each one a fractal measure, appear to dance around on the superfractal set of measures.*

By Theorem 5 there is a continuous mapping $\tilde{\mathcal{F}}_V : \Sigma_V \to \tilde{\mathfrak{H}}_V$ that assigns to each address in the code space Σ_V a V-tuple of measures in $\tilde{\mathfrak{H}}_V$. But this mappings is not helpful for characterizing $\tilde{\mathfrak{H}}_V$ because $\tilde{\mathcal{F}}_V : \Sigma_V \to \tilde{\mathfrak{H}}_V$ is not in general one-to-one, for the same reason that $\Phi : \Sigma_V \to \Omega_V$ is not one-to-one, as exhibited in Example 5.

Theorem 21. *Let $\Omega_{V,v}$ denote the set of v^{th} components of members of Ω_V. Let $\tilde{\mathcal{F}} : \Omega \to \mathbb{P}(\mathbb{X})$ be the mapping introduced in Theorem ??. Then*

$$\tilde{\mathcal{F}}(\Omega_{V,1}) = \tilde{\mathfrak{H}}_{V,1}. \tag{3.47}$$

When the probabilities in the superIFS $\mathcal{F}$ are given by (3.20) we have

$$\tilde{\mathcal{F}}(\mu_{V,1}) = \tilde{\mathfrak{P}}_{V,1} \tag{3.48}$$

where $\mu_{V,1}$ is the marginal probability distribution given by (3.25).

Definition 10. *The code tree $\Phi(a_1a_2a_3...)$ is called a tree address of the V-variable fractal measure $\tilde{\mathcal{F}}_V(a_1a_2a_3...)$.*

The mapping $\tilde{\mathcal{F}} : \Omega_{V,1} \to \mathfrak{H}_{V,1}$ together with Theorem 12 allows us to characterize V-variable fractals as follows:

Theorem 22. *Any V-variable fractal measure is made of at most V "forms" or "shapes" at each level.*

Example 9. *Example of the Theorem, in the form of a greyscale picture, pointing out the key fact!*

Theorem 23. *The set of V-variable fractal measures associated with the superIFS $\mathcal{F}$ converges to the set of random fractal measures introduced in Section 3.2; that is, in the metric of $\mathbb{H}(\mathbb{M}(\mathbb{X}))$*

$$\lim_{V\to\infty} \tilde{\mathfrak{H}}_{V,1} = \tilde{\mathfrak{H}}. \tag{3.49}$$

Moreover, when the probabilities $\{P^a | a \in \mathcal{A}\}$ obey (3.20),

$$\lim_{V\to\infty} \tilde{\mathfrak{P}}_{V,1} = \tilde{\mathfrak{P}}, \tag{3.50}$$

where the convergence is in the metric of $\mathbb{P}(\mathbb{M}(\mathbb{X}))$, and where $\tilde{\mathfrak{P}}$ is the stationary measure on random fractal measures associated with the superIFS $\mathcal{F}$.

Proof. This follows at once from Theorems 7 and 11. □

Theorem 24. *** Fractal dimension theorem, say in the case of similitudes.*

4. Algorithms, Applications and Examples

Here we consider some of the theory and applications..

4.1. Applications.

References

[1] Barnsley, M.F., "Fractals Everywhere", Academic Press, New York, (1988).
[2] Barnsley, M.F., and Hurd, L.P., "Fractal Image Compression", AK Peters, Boston, (1993)
[3] Barnsley, M.F., *Iterated Function Systems for Lossless Data Compression*, in "Fractals in Multimedia", M.F. Barnsley, D.Saupe, E.R. Vrscay (Eds.), Springer, New York, (2002).
[4] Mauldin, R.D. and Williams, S.C., *Random Recursive Constructions; Asymptotic Geometrical and Topological Properties*, Trans. Amer. Math. Soc. 295(1986), 325-346.
[5] Hutchinson, J.E., *Fractals and Self-Similarity*, Indiana. Univ. Math. J. 30 (1981), 713-749
[6] Hutchinson, J.E., and Rüschendorf, L., *Selfsimilar Fractals and Selfsimilar Random Fractals*, Progress in Probability 46(2000),109-123
[7] Hutchinson, J.E., and Rüschendorf, L.,*Random Fractal Measures via the Contraction Method*, Indiana University Mathematics Journal, 47 (1998), No. 2, 471-487.
[8] Hutchinson, J.E., and Rüschendorf, L., *Random Fractals and Probability Metrics*, Adv. Appl. Prob. (SGSA) 32(2000),925-947
[9] Hutchinson, J.E., *Deterministic and Random Fractals*, In "Complex Systems" eds. T.R.J. Bossomaier and D.G. Green, Cambridge University Press, (2000).
[10] Elton, J. *An Ergodic Theorem for Iterated Maps*, Ergod. Theory and Dynamical Systems,7(1987), 481-488.
[11] Barnsley, M.F., and Elton, J.H., *A new class of Markov processes for image encoding*, Adv. Appl. Prob.20 (1988), 14-32.
[12] Barnsley, M.F. and Demko, S., *Iterated Function Systems and the Global Construction of Fractals*, Proc. R. Soc. London Ser. A, 399 (1985), 243-275.
[13] Billingsley, P., "Convergence of Probability Measures", John Wiley, New York, (1968).
[14] Billingsley, P., "Information Theory", (**).
[15] Diaconis and Freedman (**) Siam Review Article.
[16] Falconer, K.J., *Random Fractals*, Math. Proc. Camb. Phil. Soc. 100(1986), 559-582.
[17] Graf, S., *Statistically Self-Similar Fractals*, Prob. Theor. Rel. Fields 74(1987), 357-392.
[18] Graf, S., Mauldin, R.D. and Williams, S.C., *The Exact Hausdorff Dimension in Random Recursive*, Mem. Amer. Math. Soc. 71(1988), No. 381.
[19] Katok, A. and Hasselblatt, B.,"Introduction to the Modern Theory of Dynamical Systems", Cambridge University Press, 1995.
[20] Saupe, D., *Algorithms for Random Fractals*, In "The Science of Fractal Images", eds. H.O. Peitgen and D. Saupe, Springer, New York, pp.71-100, (1988)
[21] Arbeiter, M.A.(1991) *Random Recursive Constructions of Self-Similar Fractal Measures, the Noncompact Case*, Prob. Theor. Rel. Fields 88(1991), 497-520.
[22] Kahane, J.P. and Peyrière, J., *Sur Certaines Martingales de Mandelbrot*, Adv. Math 22(1976), 131-145.
[23] Olsen, L., *Random Geometrically Graph Directed Self-Similar Multifractals*, (Pitman Research Notes 307) (1994) Longman, Harlow
[24] Patzschke, N. and Zähle, U. *Self-Similar Random Measures. IV. The Recursive Construction Model of Falconer, Graf, and Mauldin and Williams*, Math. Nachr. 149(1990), 285-302.
[25] Zähle, U. *Self-Similar Random Measures. I. Notion, Carrying Hausdorff Dimension, and Hyperbolic Distribution* Prob. Theor. Rel. Fields 80 (1988), 79-100.

335 Pennbrooke Trace, Duluth,, GA 30097, USA
E-mail address: Mbarnsley@aol.com

Department of Mathematics, Australian National University
E-mail address: john.hutchinson@anu.edu.au

Department of Mathematics, Umea University, SE-90187 Umea, Sweden
E-mail address: stenflo@math.umu.se

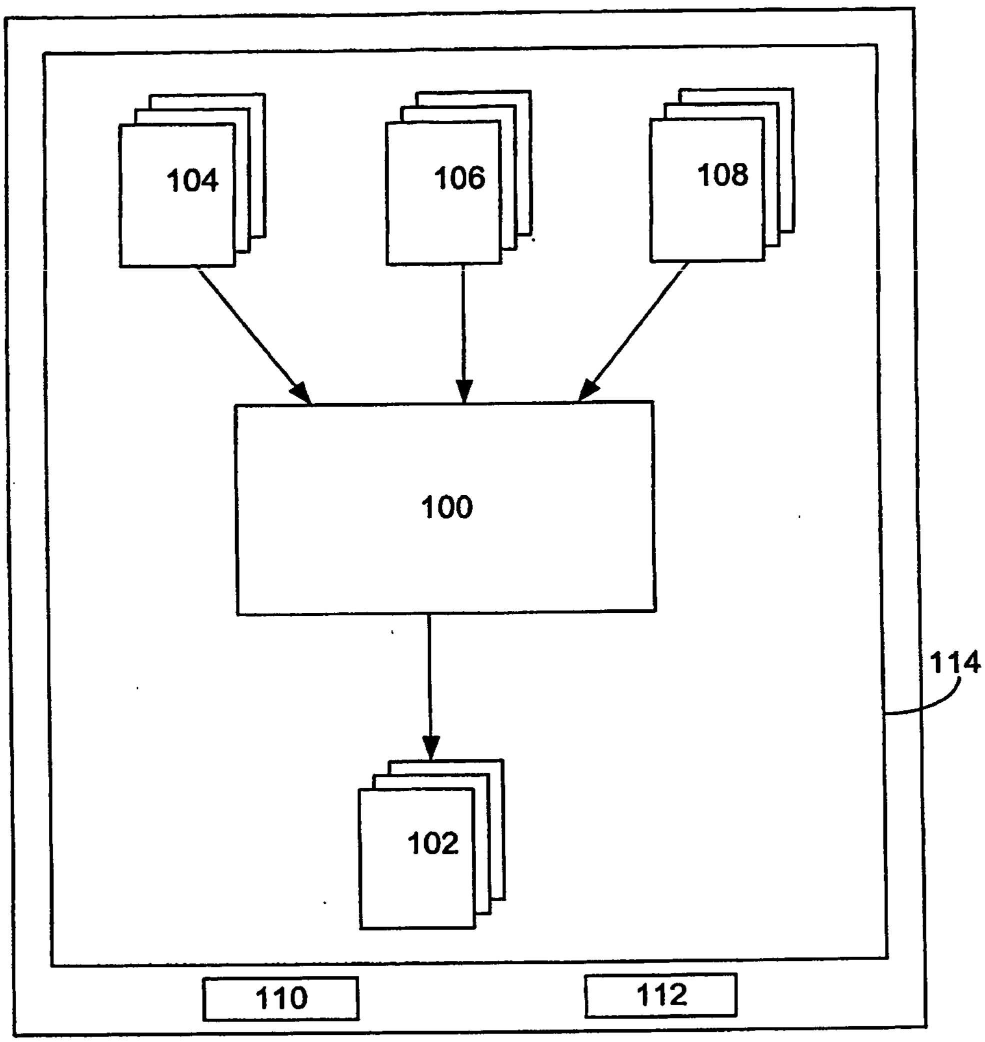


Figure 1

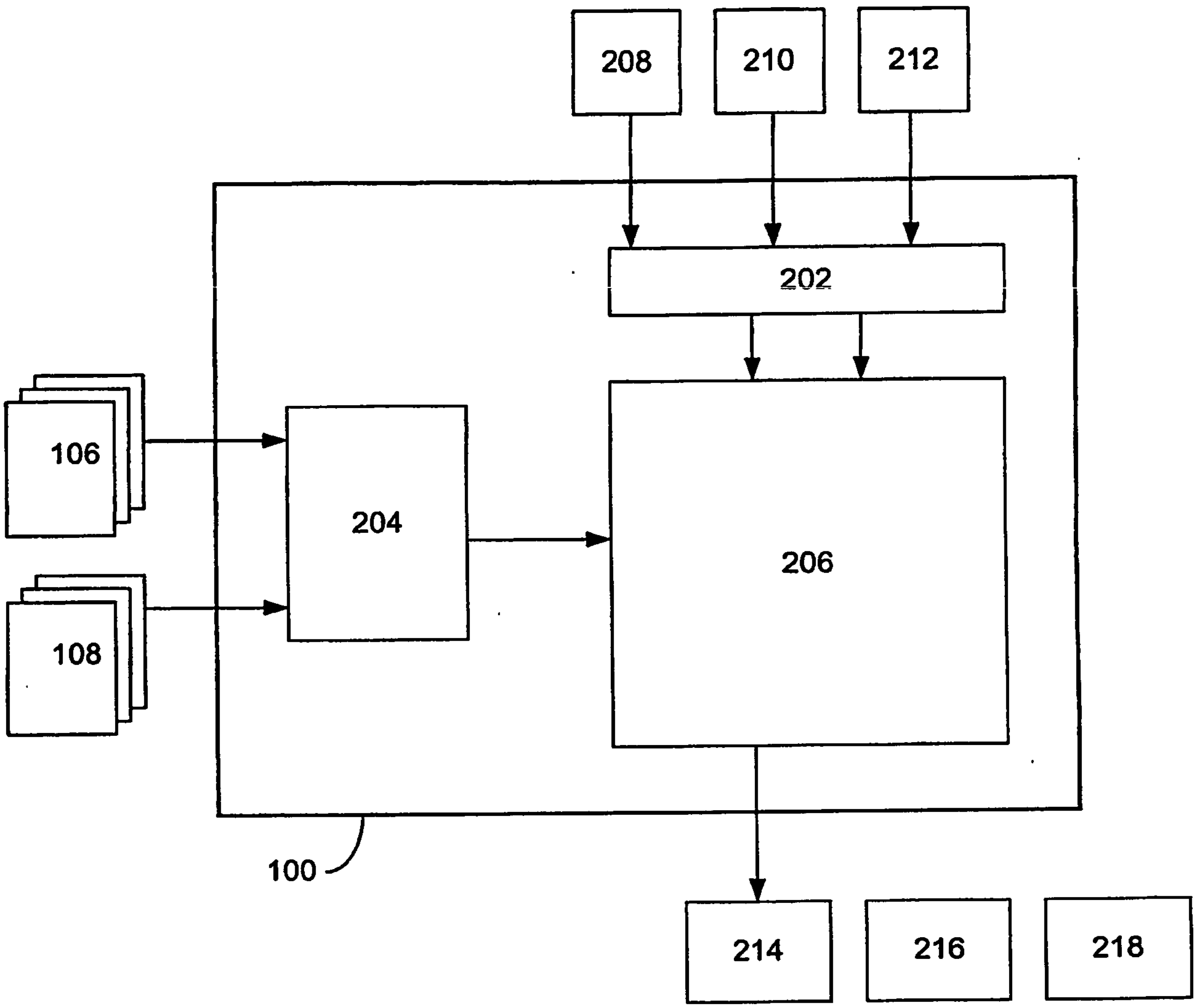


Figure 2

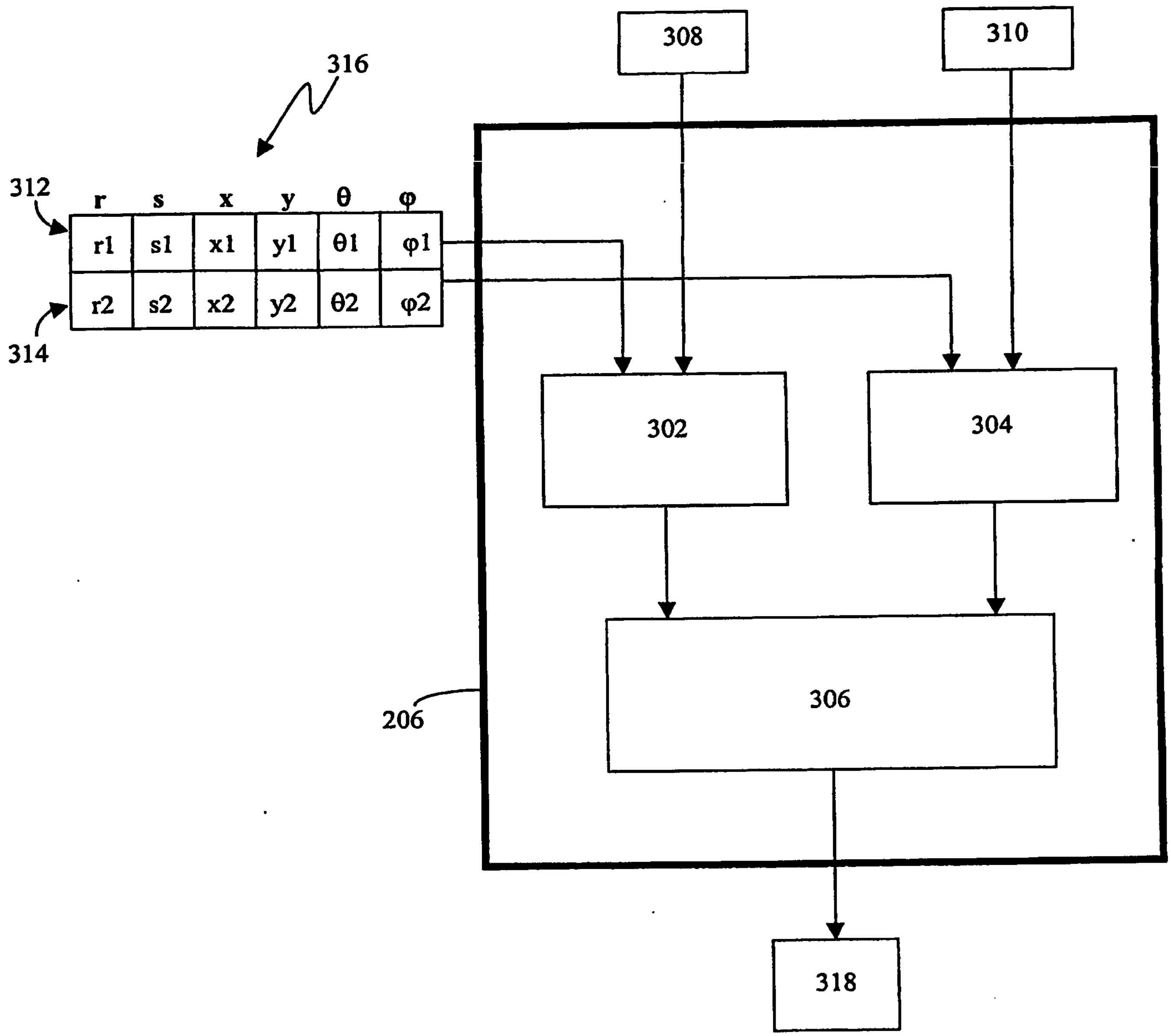
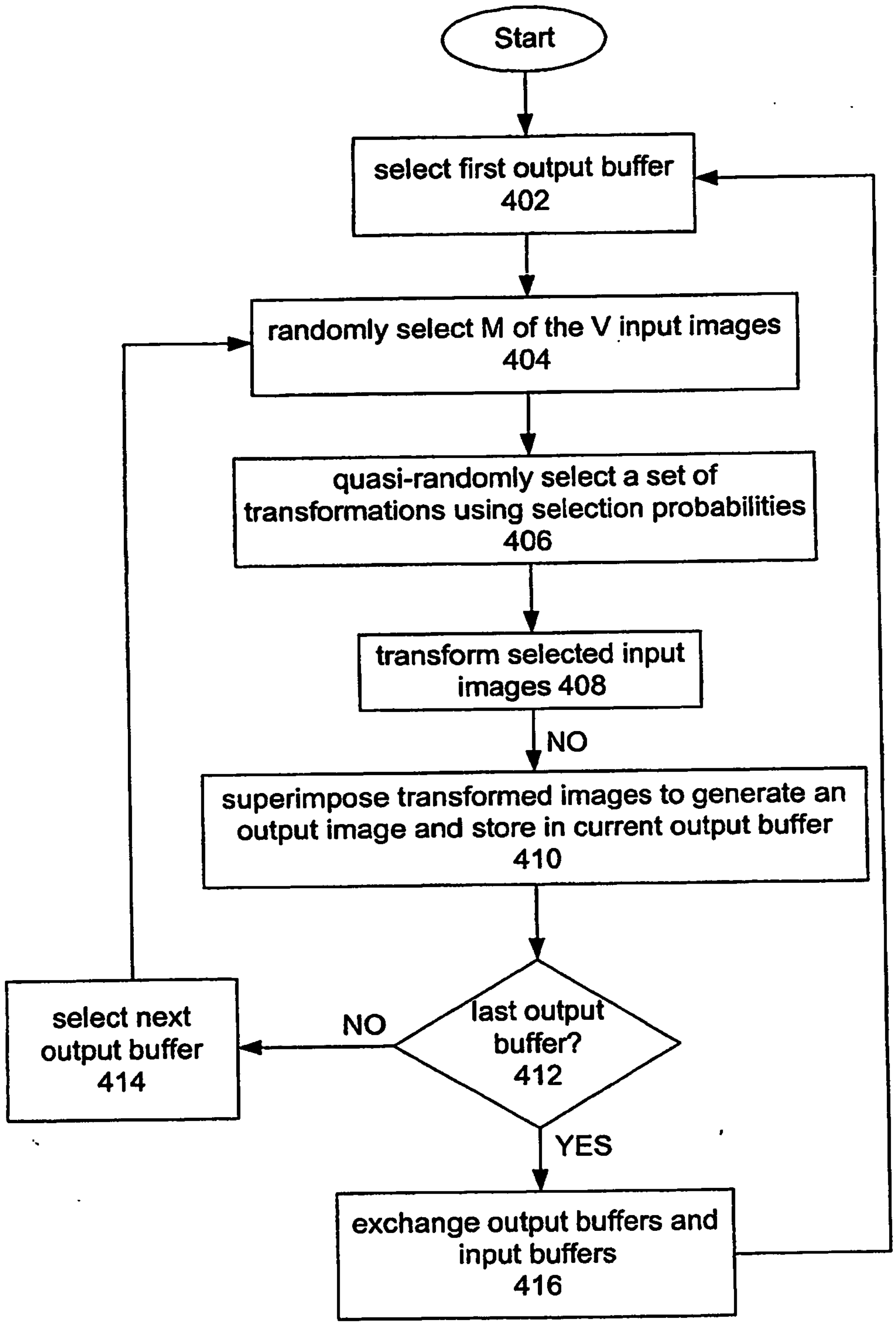
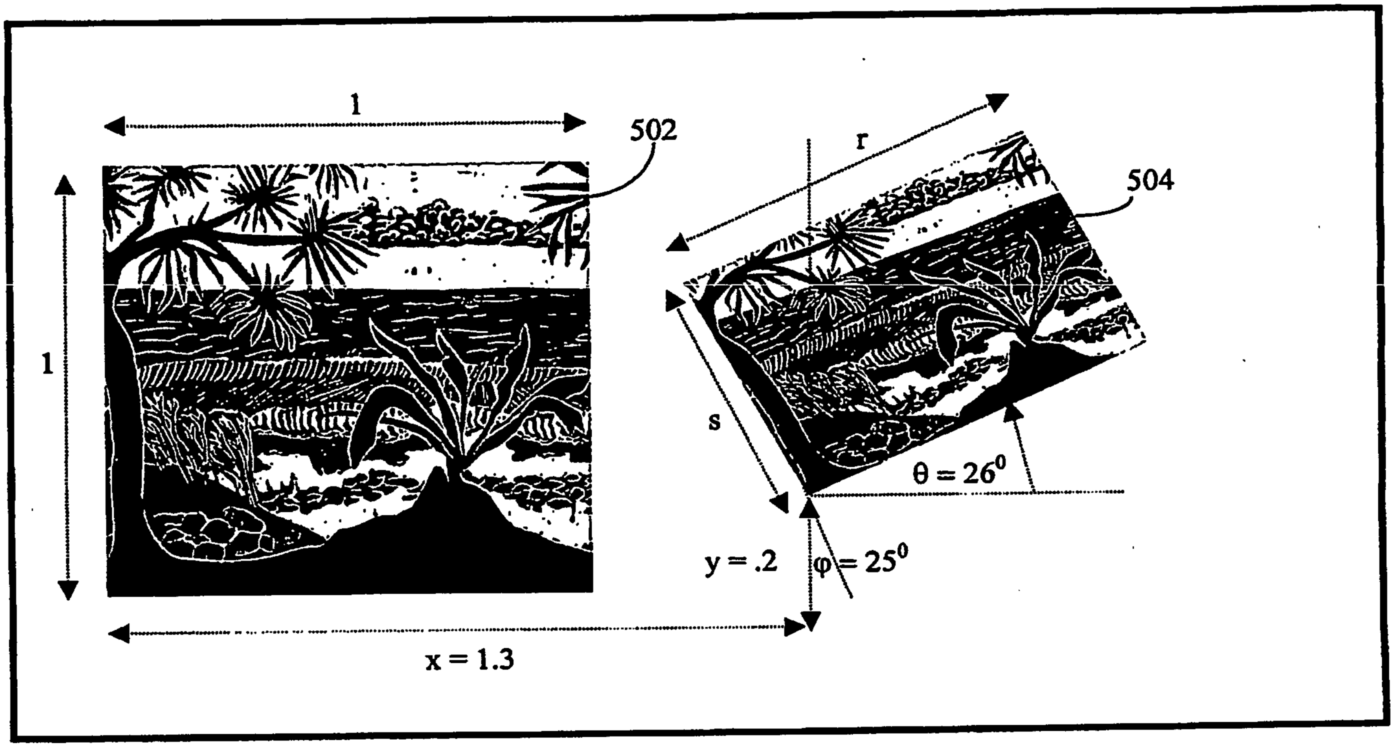

316
312
r s x y θ φ
r1 s1 x1 y1 θ1 φ1
r2 s2 x2 y2 θ2 φ2
314
308
310
302
304
306
206
318

Figure 3



Start
select first output buffer 402
randomly select M of the V input images 404
quasi-randomly select a set of transformations using selection probabilities 406
transform selected input images 408
NO
superimpose transformed images to generate an output image and store in current output buffer 410
last output buffer? 412
NO
select next output buffer 414
YES
exchange output buffers and input buffers 416

Figure 4



r	s	x	y	θ	φ
.9	.6	1.3	.2	26^0	25^0

Figure 5

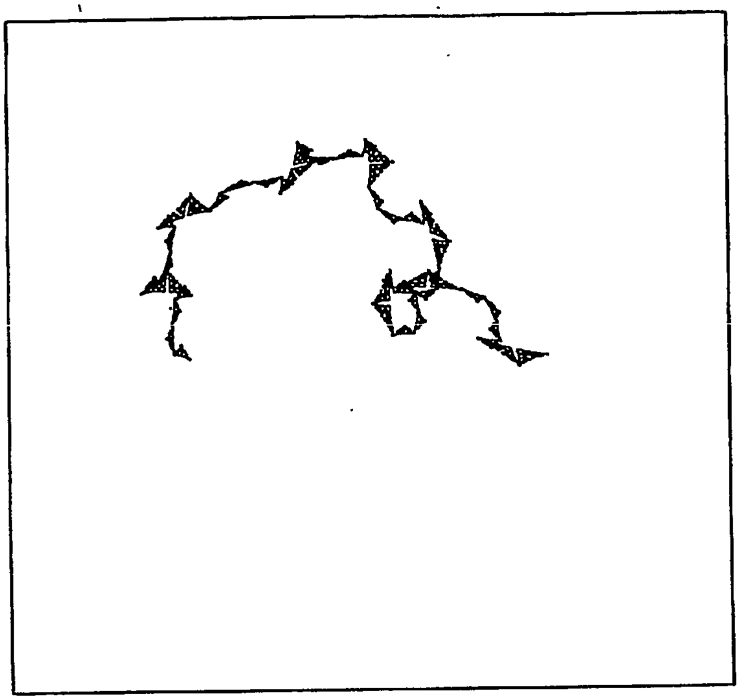


Figure 6

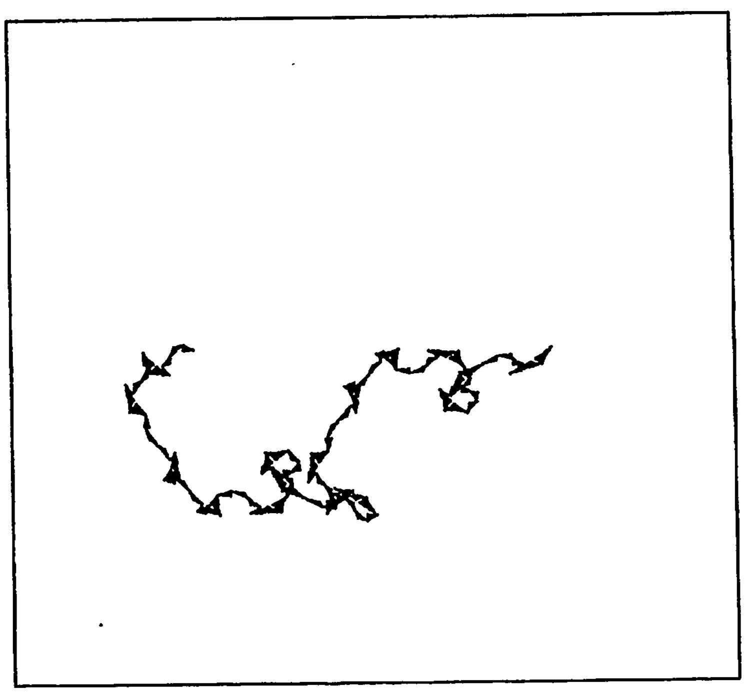


Figure 7

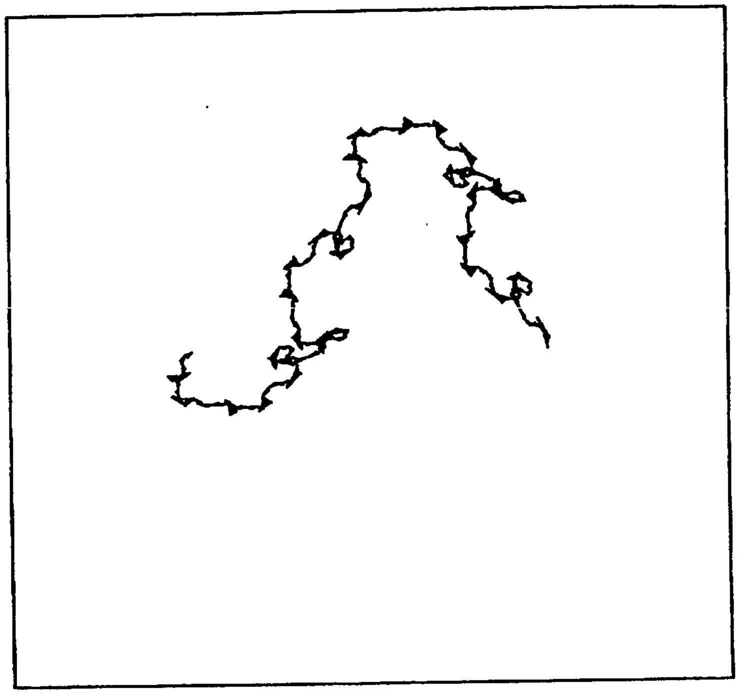


Figure 8

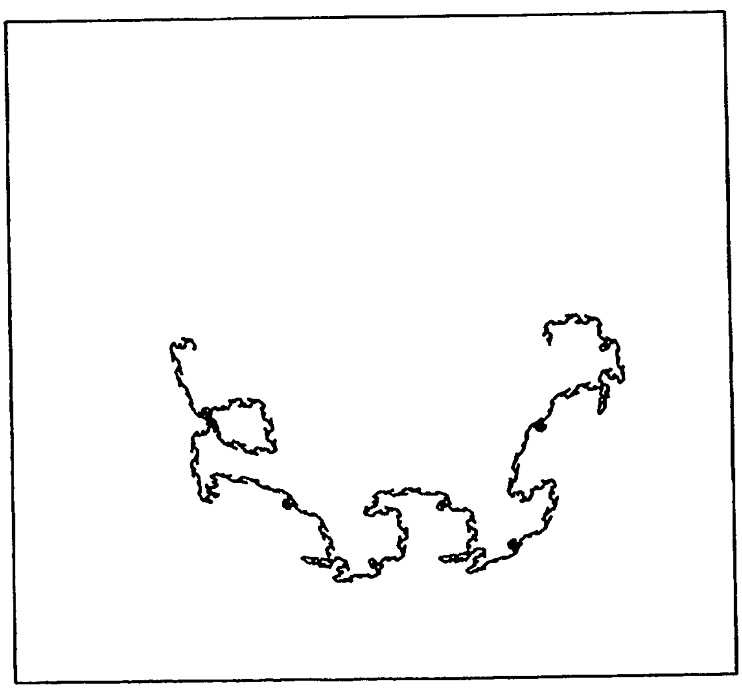


Figure 9

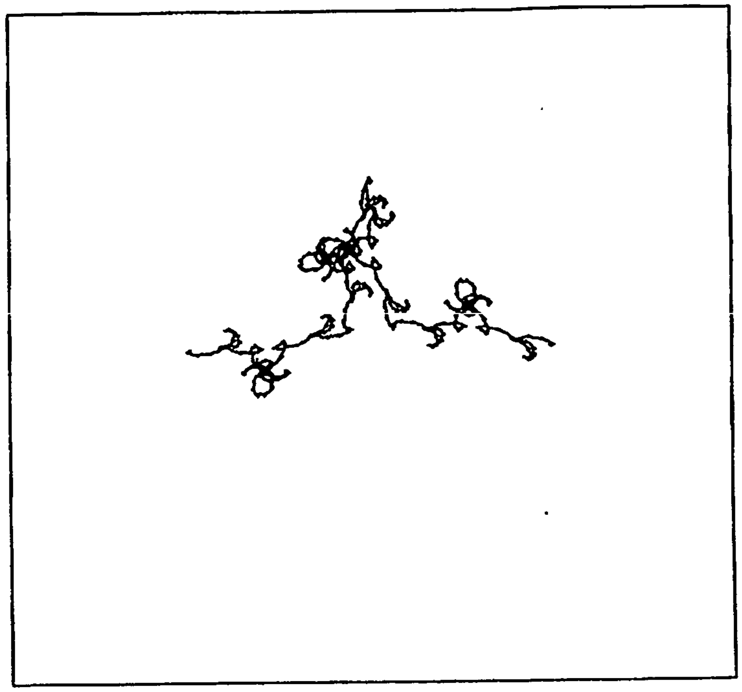


Figure 10

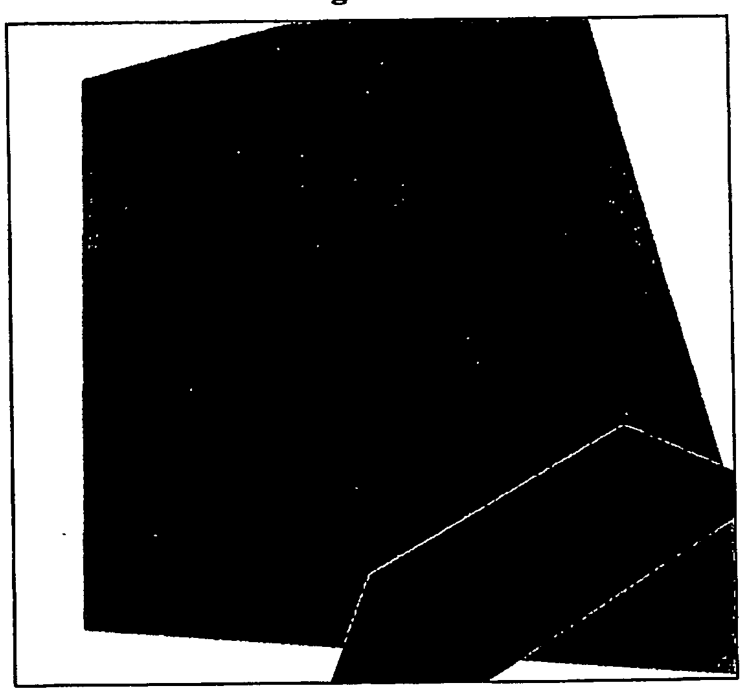


Figure 11

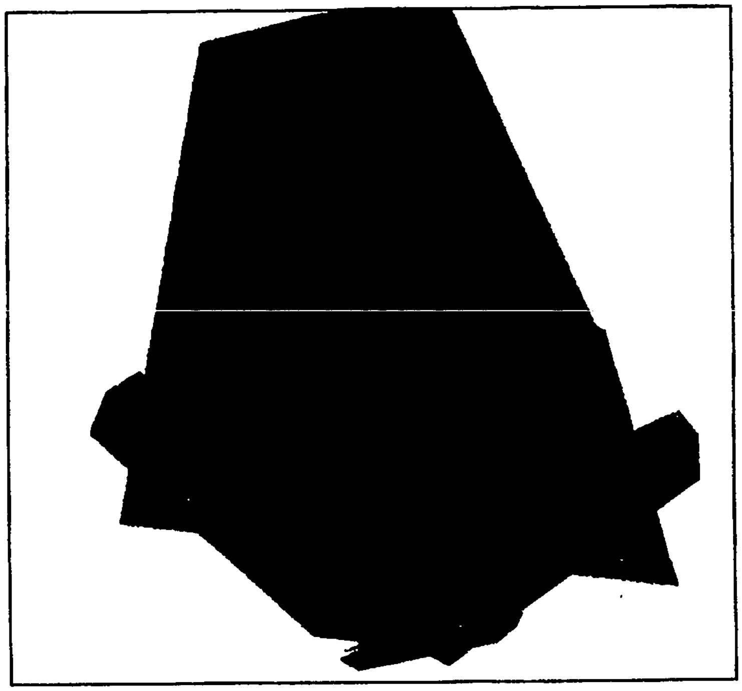


Figure 12

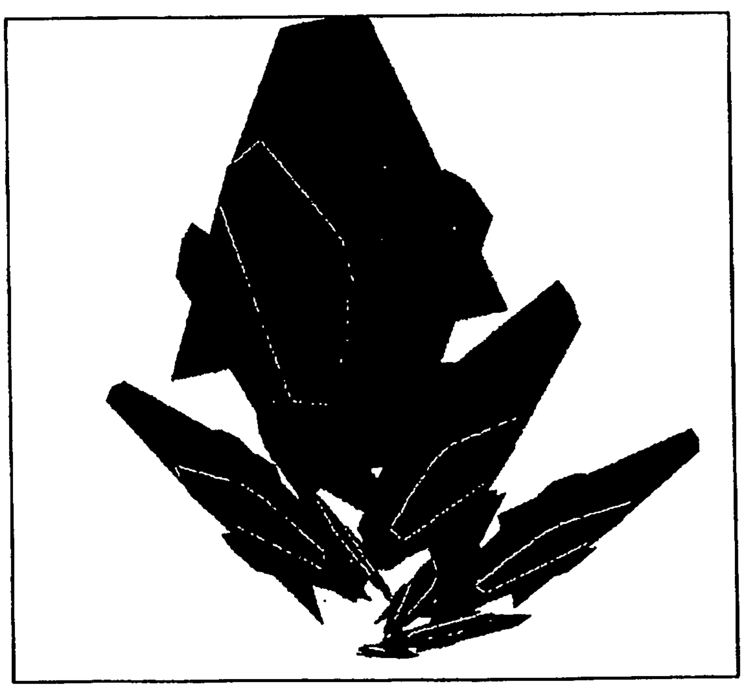


Figure 13

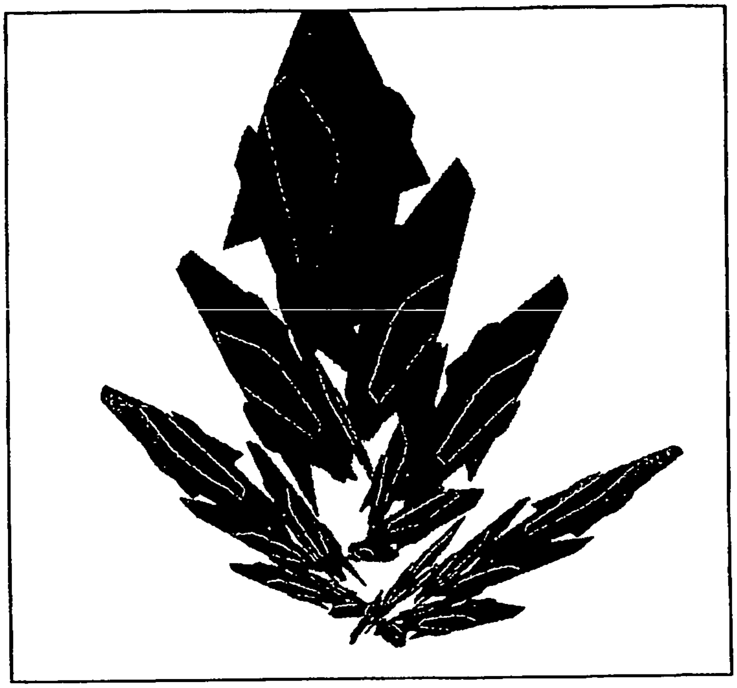


Figure 14

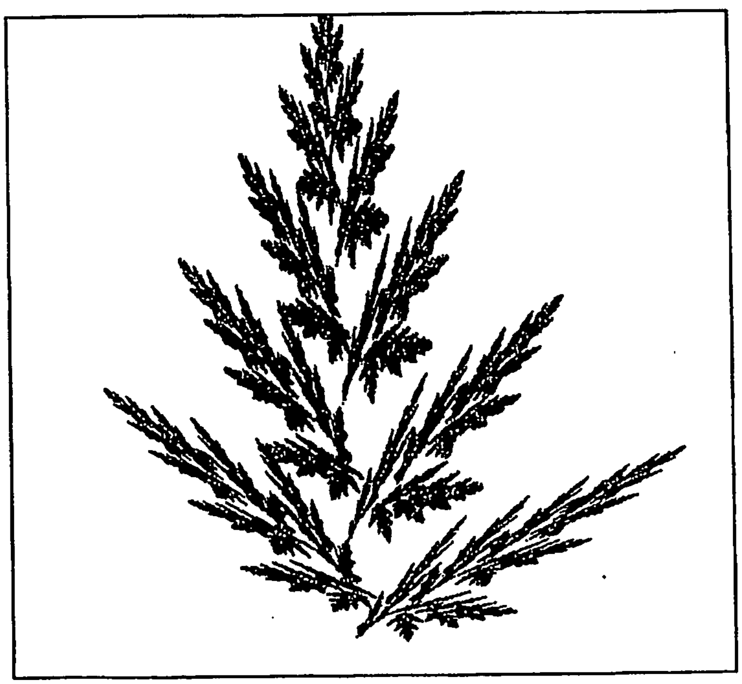


Figure 15

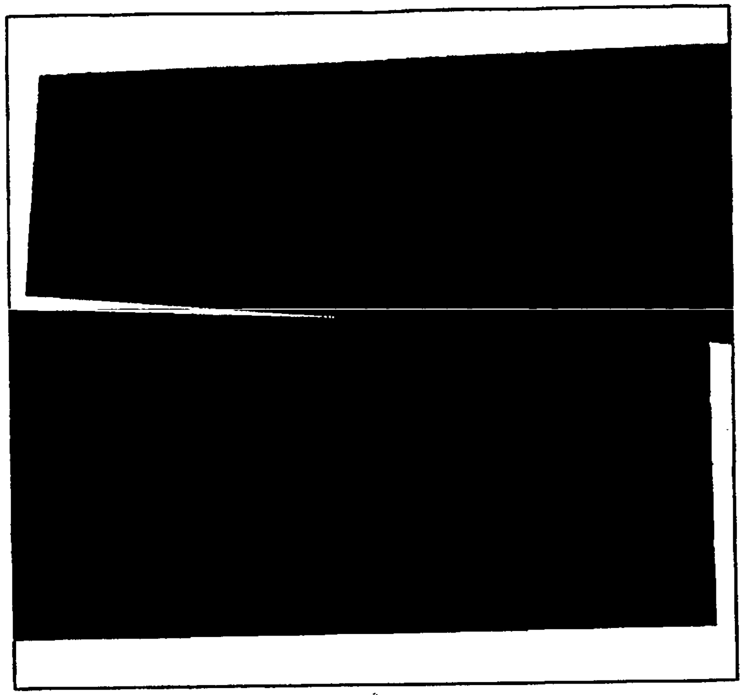


Figure 16

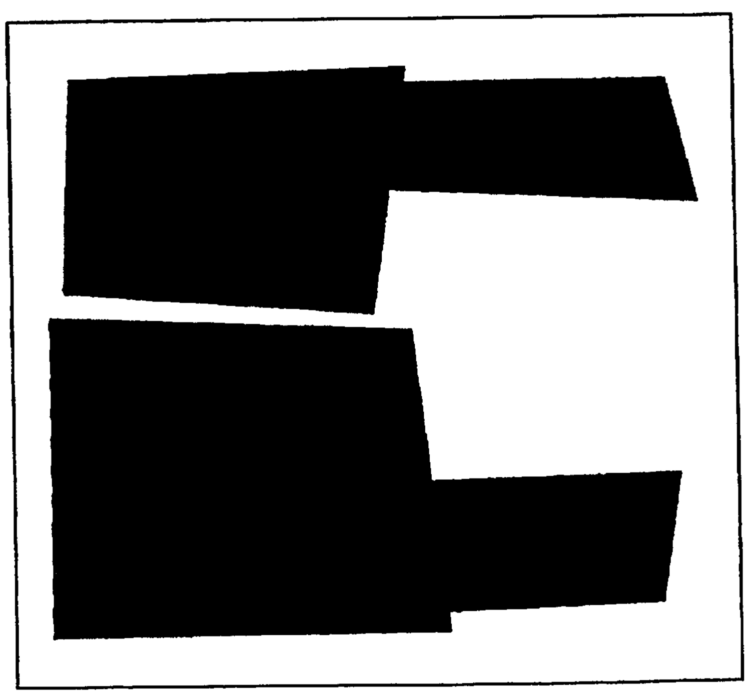


Figure 17

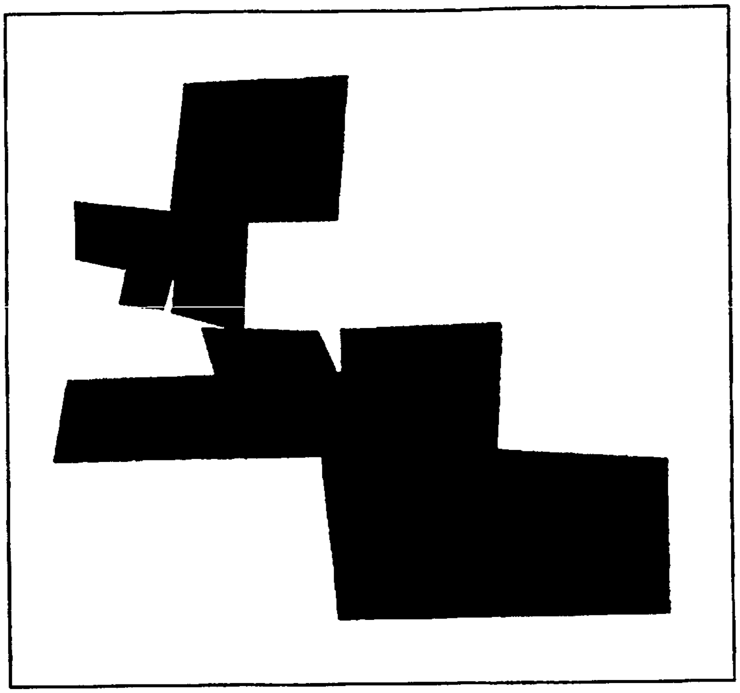


Figure 18

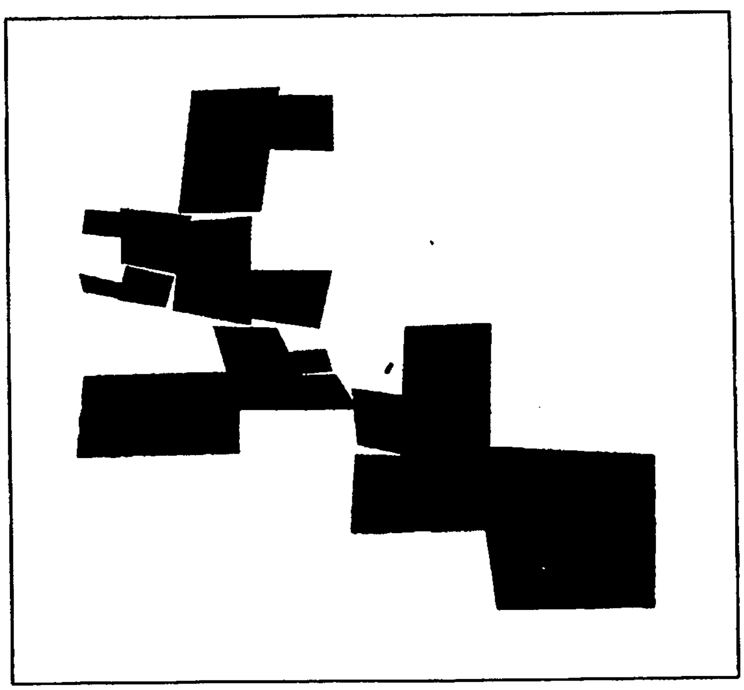


Figure 19

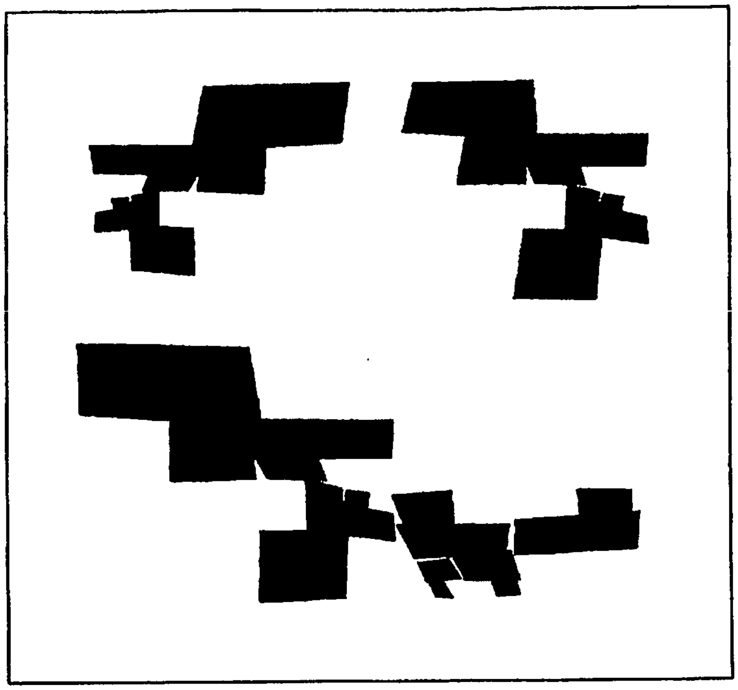


Figure 20

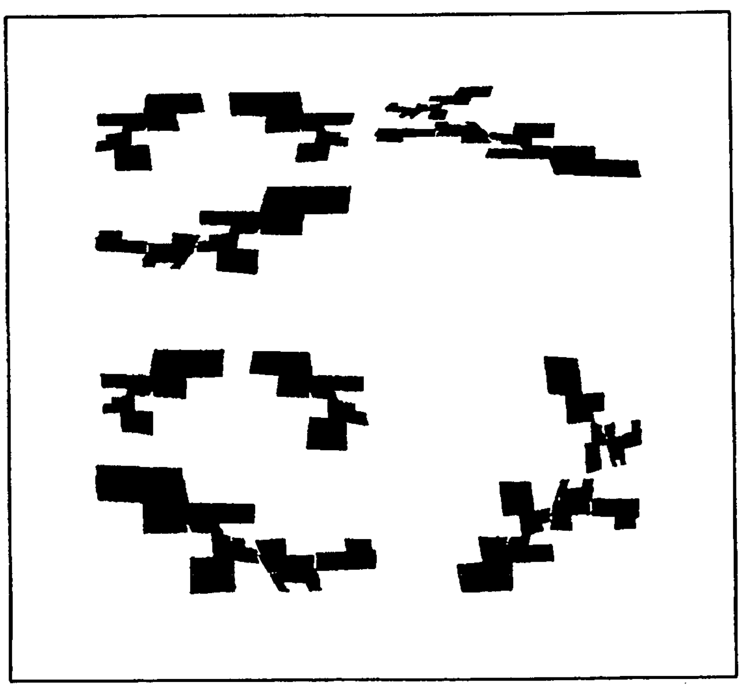


Figure 21

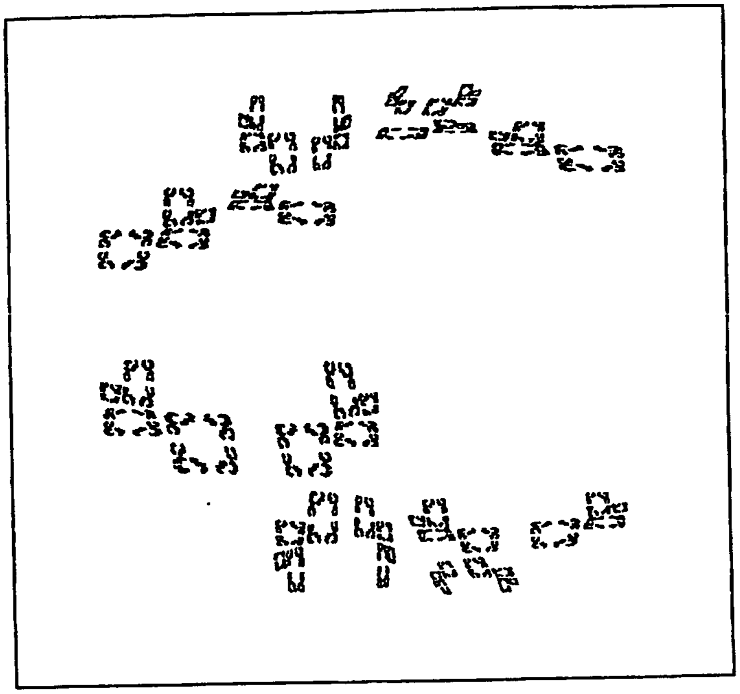


Figure 22

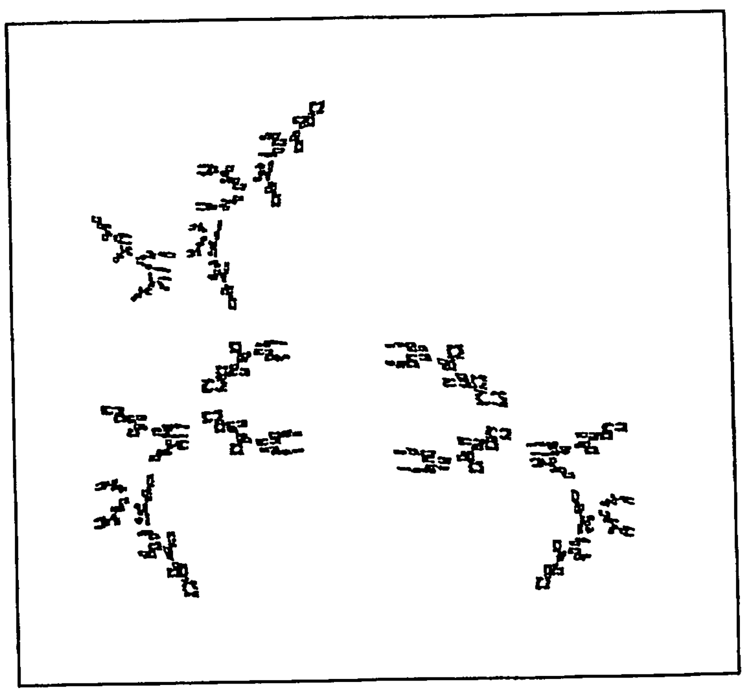


Figure 23

This Page is inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ BLACK BORDERS
- ☒ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☒ BLURED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLORED OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REPERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: ______________________________

IMAGES ARE BEST AVAILABLE COPY.
As rescanning documents *will not* correct images problems checked, please do not report the problems to the IFW Image Problem Mailbox